www.linuxuser.co.uk

Linux User & Developer™

THE ESSENTIAL MAGAZINE
FOR THE GNU GENERATION

WORK FASTER, MORE SECURELY, FROM ANYWHERE

TAKE CONTROL OF CONTAINERS

• Speed up system resources • Run Linux on any machine • Deploy software on demand

35 ESSENTIAL LINUX FIXES

Troubleshoot problems and improve performance with these pro tips

RESTRICT UNTRUSTED SOFTWARE WITH FIREJAIL

Reduce security risks by running apps in a sandbox

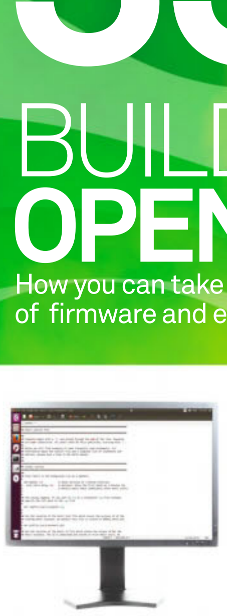
DISC MISSING? ASK YOUR RETAILER

BUILD YOUR OWN OPENWRT IMAGE

How you can take complete customised control of firmware and embedded systems on the Pi

KEEP YOUR DATA SAFE

RUN A VPN THROUGH YOUR PI



CONFIGURE A LAMP STACK

Combine Linux, Apache, MySQL and PHP for the ultimate web platform

ADMINISTRATE UBUNTU SERVERS

Improve your sysadmin skills

ALSO INSIDE

» Arya Linux on test
» Program with Erlang
» Add notifications with Blinkt

Future
ISSUE 176
PRINTED IN THE UK £6.49
Digital Edition
GreatDigitalMags.com

Synology®

DS216play | DS716+II | DS216+II

4K TRANSCODING & STORAGE SOLUTION

Ultra HD 4K dlna CERTIFIED AirPlay

4K VIDEO TRANSCODING

Transcode 4K videos to make them suitable for all screens and bandwidths whether they are local media players or remote mobile devices.

TURN YOUR NAS INTO A MEDIA CENTRE

Access and share all the multimedia files stored on your Synology NAS anytime, anywhere for home or work through our large range of apps.

STORE AND STREAM YOUR MEDIA

Stream digital content with Roku, Apple TV, Chromecast, Samsung TV and all Bluetooth, Airplay and DLNA devices effortlessly.

For where to buy visit www.synology.com

YouTube

Available mobile Apps

iOS

Synology's Media Accolades

PCR awards 2016 WINNER

PC PRO BEST NAS BRAND 2016 SYNOLOGY WINNER

LinuxUser & Developer
THE MAGAZINE FOR
THE GNU GENERATION

Future Publishing Ltd
Richmond House, 33 Richmond Hill
Bournemouth, Dorset, BH2 6EZ
☎ +44 (0) 1202 586200
Web: **www.linuxuser.co.uk**
www.greatdigitalmags.com
www.futureplc.com

Editorial
Editor **April Madden**
april.madden@futurenet.com
☎ 01202 586218
Senior Art Editor **Stephen Williams**
Designer **Rebekka Hearl**
Editor in Chief **Paul Newman**
Photographer **James Sheppard**

Contributors
Dan Aldred, Joey Bernard, Nate Drake, Tam Hanna, Oliver Hill, Phil King, Kushma Kumari, Paul O'Brien, Swayam Prakasha, Eleanor Rewcastle, Richard Smedley, Nicholas Thorpe, Nitish Tiwari and Mihalis Tsoukalos

Advertising
Digital or printed media packs are available on request.
Commercial Sales Director **Clare Dove**
Advertising Director **Richard Hemmings**
richard.hemmings@futurenet.com
☎01225 687615
Account Director **Andrew Tilbury**
andrew.tilbury@futurenet.com
☎01225 687144
Account Director **Crispin Moller**
crispin.moller@futurenet.com
☎01225 687335

International
Linux User & Developer is available for licensing. Contact the International department to discuss partnership opportunities.
Head of International Licensing **Cathy Blackman**
☎ +44 (0) 1202 586401
cathy.blackman@futurenet.com

Subscriptions
For all subscription enquiries:
LUD@servicehelpline.co.uk
☎ 0844 249 0282
☎ Overseas +44 (0)1795 418661
www.imaginesubs.co.uk
Head of subscriptions **Sharon Todd**

Circulation
Circulation Director **Darren Pearce**
☎ 01202 586200

Production
Production Controller **Nola Cokely**

Management
Finance & Operations Director **Marco Peroni**
Creative Director **Aaron Asadi**
Editorial Director **Ross Andrews**

Printing & Distribution
William Gibbons, 26 Planetary Road, Willenhall, West Midlands, WV13 3XT

Distributed in the UK, Eire & the Rest of the World by
Marketforce, 5 Churchill Place, Canary Wharf, London, E14 5HU
☎ 0203 787 9060 www.marketforce.co.uk

Distributed in Australia by Gordon & Gotch Australia Pty Ltd,
26 Rodborough Road, Frenchs Forest, New South Wales 2086
☎ + 61 2 9972 8800 www.gordongotch.com.au

Disclaimer

The publisher cannot accept responsibility for any unsolicited material lost or damaged in the post. All text and layout is the copyright of Future Publishing Ltd. Nothing in this magazine may be reproduced in whole or part without the written permission of the publisher. All copyrights are recognised and used specifically for the purpose of criticism and review. Although the magazine has endeavoured to ensure all information is correct at time of print, prices and availability may change. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to Future Publishing via post, email, social network or any other means, you automatically grant Future Publishing an irrevocable, perpetual, royalty-free licence to use the material across its entire portfolio, in print, online and digital, and to deliver the material to existing and future clients, including but not limited to international licensees for reproduction in international, licensed editions of Future Publishing products. Any material you submit is sent at your risk and, although every care is taken, neither Future Publishing nor its employees, agents or subcontractors shall be liable for the loss or damage.

ISSN 2041-3270

recycle
When you have finished with this magazine please recycle it.

Future

Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.

Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR).
www.futureplc.com

Chief executive Zillah Byng-Thorne
Non-executive chairman Peter Allen
Chief financial officer Penny Ladkin-Brand

Tel +44 (0)1225 442 244

Welcome

to issue 176 of Linux User & Developer

This issue

» Take control of containers
» 35 essential Linux fixes
» Restrict untrusted software with Firejail
» Run a VPN through your Pi

Welcome to the latest issue of Linux User & Developer, the UK and America's favourite Linux and open source magazine. Have you joined the container revolution yet? Whether you're a developer looking for a complete runtime environment, a sysadmin looking for secure, locked down infrastructure or you simply want to explore the latest development in computing, our feature on p18 will explain everything you need to know, from the principles of containerisation to the key tools and how to use them.

Also this issue, discover how to solve 35 of the most annoying Linux problems that you could encounter. Whether you're new to Linux or you're a seasoned pro, these particular issues have a habit of raising their ugly heads. We explain how to fix all of them on p56. Plus, learn how to set up a LAMP stack, sandbox an application using Firejail, continue working with Erlang and OpenWRT, take a look at systems administration under Ubuntu, check out the best password managers, the newest Synology router, the latest version of AryaLinux and much more. Our Practical Raspberry Pi section also has some great guides to setting up a VPN and getting to grips with the Pimoroni Blinkt! board, and we take a look at a fantastic Pi project for the green-fingered among you.

Enjoy the issue!
April Madden, Editor

Get in touch with the team:
linuxuser@imagine-publishing.co.uk

Facebook: Linux User & Developer
Twitter: @linuxusermag
Buy online imagineshop.co.uk

Visit us online for more news, opinion, tutorials and reviews:
www.**linuxuser**.co.uk

LinuxUser & Developer

Contents

Subscribe & save!

32 Check out our great new offer! US customers can subscribe on page 70

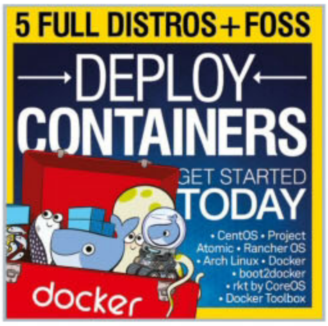
18 Take control of containers
Containers are revolutionising DevOps and infrastructure

OpenSource

08 News
The biggest stories from the open source world

12 Interview
Pavel Baiborodin on the Blynk platform

16 Kernel column
The latest on the Linux kernel with Jon Masters



5 FULL DISTROS+FOSS

DEPLOY CONTAINERS

GET STARTED TODAY

• CentOS • Project Atomic • Rancher OS • Arch Linux • Docker • boot2docker • rkt by CoreOS • Docker Toolbox

Features

18 Take control of containers
Revolutionise your DevOps

56 35 essential fixes
Learn how to solve the most annoying Linux problems

Tutorials

34 Create a custom image of OpenWRT
Take in-depth control of your router's firmware

38 Configure a LAMP stack
Deploy your own websites with ease on top of a Linux-Apache-MySQL-PHP stack

42 Sandbox applications with Firejail
Run a potentially insecure application in a sandbox to protect your system and data

46 Program in Erlang: File I/O
Discover how to work with file operations in Erlang

52 Server administration in Ubuntu
Take a closer look at server admininstration under Ubuntu

FileSilo

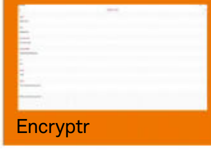
96 Free downloads
Find out what we've uploaded to our secure repo FileSilo for you this month

Reviews

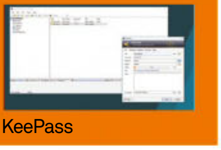
83 Password managers
Which one of these FOSS picks will keep your logins locked down?

Clipperz

Passopolis



Encryptr

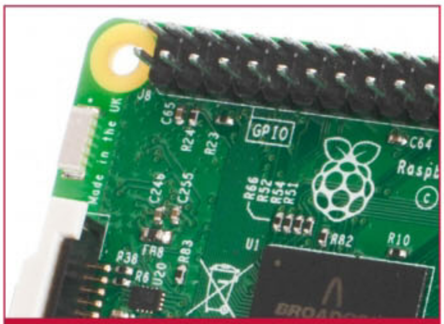


KeePass

88 Synology RT2600ac
The latest Synology router is pricey, but is that justified? Find out

90 AryaLinux 2017
Can a distro usurp Puppy Linux as king of lightweight low-resource distros?

92 Free software
Richard Smedley recommends some excellent FOSS packages for you to try



71 Practical Raspberry Pi
Use a virtual private network with your Pi, get to grips with the Pimoroni Blinkt! board, discover how the Pi can monitor plant life and much more

Join us online for more Linux news, opinion and reviews www.linuxuser.co.uk

On your free DVD this issue

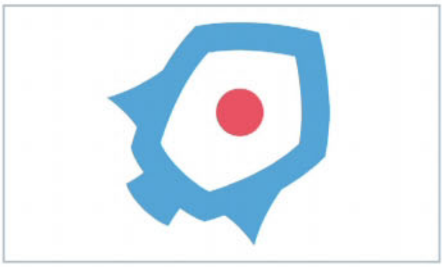
Find out what's on your free disc

Welcome to the Linux User & Developer DVD. This issue, discover some of the essential tools you need to take control of containers. From must-have FOSS like Docker to lightweight distros designed to help you work with containers or run on minimal and bare-metal systems, you'll find everything you need to get started. You can even create and manage containers on Windows-based systems too!

Featured software:

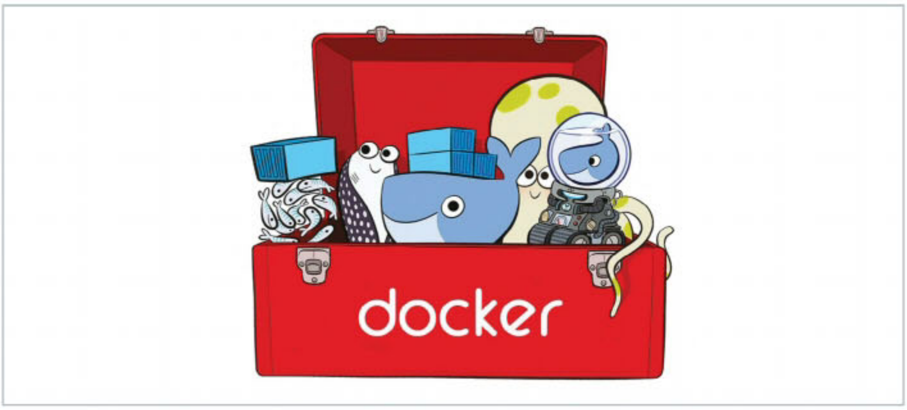
Docker

Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries – anything that can be installed on a server. Find out more about how you can use Docker and take advantage of its complete containerisation package in our complete guide on p18.



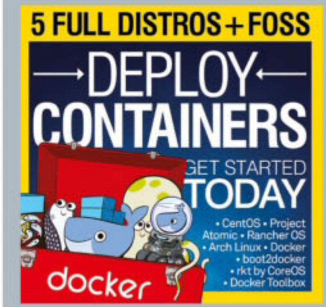
rkt by CoreOS

rkt is the next-generation container manager for Linux clusters. Designed for security, simplicity, and composability within modern cluster architectures, rkt discovers, verifies, fetches, and executes application containers with pluggable isolation. This means that you can containerise individual apps and everything that's needed to run them.



Docker Toolbox

Containers aren't limited to the Linux side of your system! The Docker Toolbox is an installer to quickly and easily install and set up a Docker environment on your computer. This is the Windows version. Using Docker Toolbox, you'll be able to set up specialist containerisation distros and environments to run on a Windows machine.



5 FULL DISTROS + FOSS

DEPLOY CONTAINERS

GET STARTED TODAY

- CentOS • Project Atomic • Rancher OS • Arch Linux • Docker • boot2docker • rkt by CoreOS • Docker Toolbox

docker

Load DVD

To access software and tutorial files, simply insert the disc into your computer and double-click the icon.

Live boot

To live-boot into the distros supplied on this disc, insert the disc into your disc drive and reboot your computer.

Please note:

- You will need to ensure that your computer is set up to boot from disc (press F9 on your computer's BIOS screen to change Boot Options).
- Some computers require you to press a key to enable booting from disc – check your manual or the manufacturer's website to find out if this is the case on your PC.
- Live-booting distros are read from the disc: they will not be installed permanently on your computer unless you choose to do so.

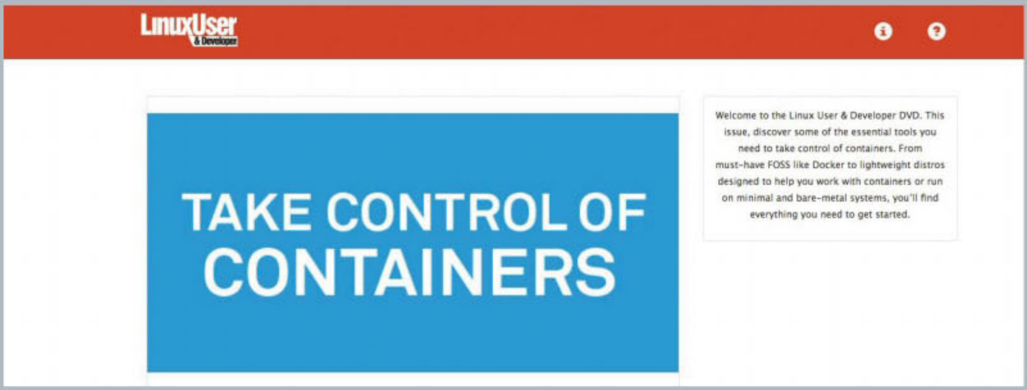
For best results:

This disc has been optimised for modern browsers capable of rendering recent updates to the HTML and CSS standards. So to get the best experience we recommend you use:

- Internet Explorer 8 or higher
- Firefox 3 or higher
- Safari 4 or higher
- Chrome 5 or higher

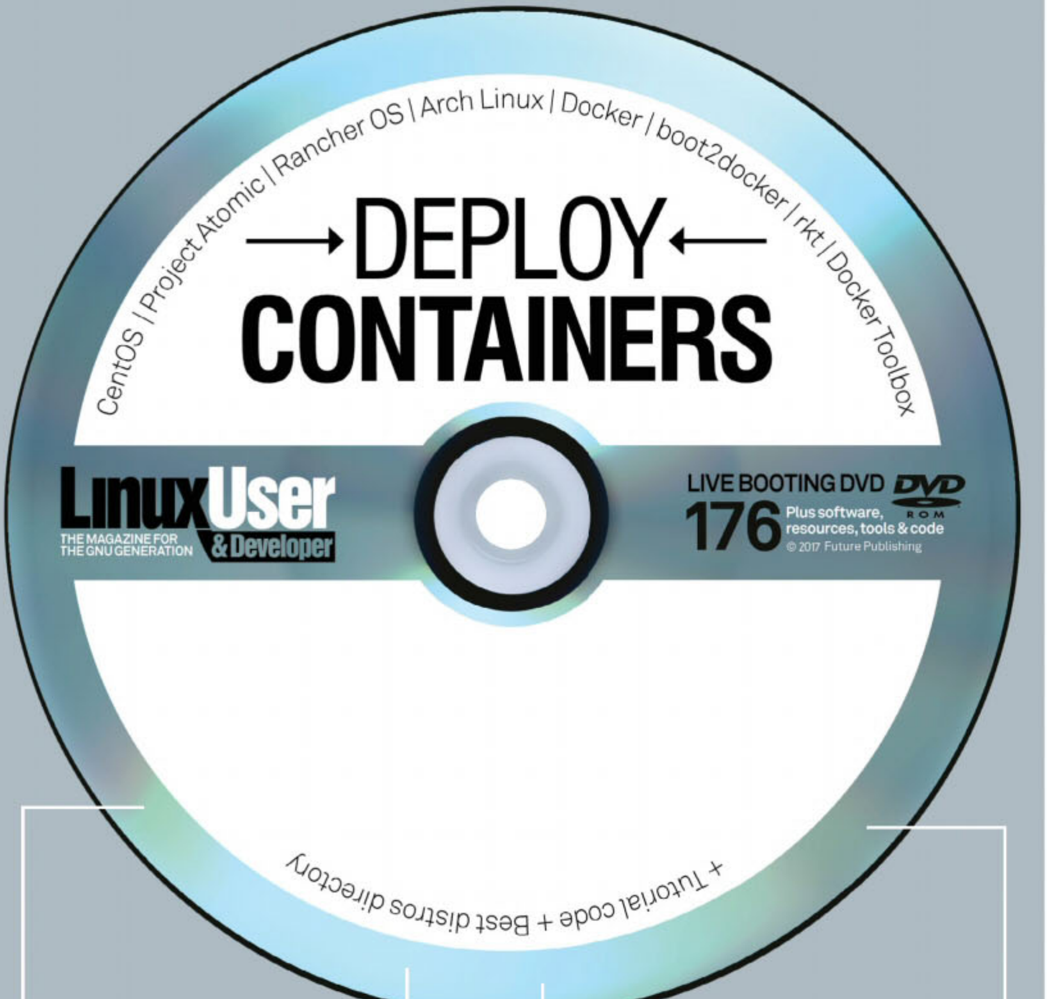
Problems with the disc?

Send us an email at **linuxuser@imagine-publishing.co.uk** Please note however that if you are having problems using the programs or resources provided, then please contact the relevant software companies.



TAKE CONTROL OF CONTAINERS

Welcome to the Linux User & Developer DVD. This issue, discover some of the essential tools you need to take control of containers. From must-have FOSS like Docker to lightweight distros designed to help you work with containers or run on minimal and bare-metal systems, you'll find everything you need to get started.


CentOS | Project Atomic | Rancher OS | Arch Linux | Docker | boot2docker | rkt | Docker Toolbox
→DEPLOY←
CONTAINERS
LinuxUser & Developer
THE MAGAZINE FOR THE GNU GENERATION
LIVE BOOTING DVD
DVD ROM
176
Plus software, resources, tools & code
© 2017 Future Publishing
+ Tutorial code + Best distros directory

Live boot

Insert the disc into your computer and reboot. You will need to make sure that your computer is set up to boot from disc

FOSS

Free and open-source software needs to be installed via the distros or by using the disc interface

Explore

Alternatively you can insert and run the disc to explore the interface and content

Distros

Distros can be live booted so that you can try a new operating system instantly without making permanent changes to your computer

Disclaimer

Important information

Check this before installing or using the disc

For the purpose of this disclaimer statement the phrase 'this disc' refers to all software and resources supplied on the disc as well as the physical disc itself.

You must agree to the following terms and conditions before using this 'this disc':

Loss of data

In no event will Future Publishing accept liability or be held responsible for any damage, disruption and/or loss to data or computer systems as a result of using 'this disc'. Future Publishing makes every effort to ensure that 'this disc' is delivered to you free from viruses and spyware. We do still strongly recommend that you run a virus checker over 'this disc' before use and that you have an up-to-date backup of your hard drive before using 'this disc'.

Hyperlinks:

Future Publishing does not accept any liability for content that may appear as a result of visiting hyperlinks published in 'this disc'. At the time of production, all hyperlinks on 'this disc' linked to the desired destination. Future Publishing cannot guarantee that at the time of use these hyperlinks direct to that same intended content as Future Publishing has no control over the content delivered on any of these hyperlinks.

Software Licensing

Software is licensed under different terms; please check that you know which one a program uses before you install it.

- **Shareware**: If you continue to use the program you should register it with the author
- **Freeware**: You can use the program free of charge
- **Trials/Demos**: These are either time-limited or have some functions/features disabled
- **Open source/GPL**: Free to use, but for more details please visit **https://opensource.org/licenses/gpl-license**

Unless otherwise stated you do not have permission to duplicate and distribute 'this disc'.

Above The KDE Slimbook comes with the fantastic KDE Neon pre-installed

HARDWARE

KDE unveils lightweight Linux laptop

Move over Macbook, there's a new king in town

For many looking to get their Linux fix, choices primarily revolve around which desktop PC you're going to buy. Laptops pre-loaded with Linux at their core are few and far between, and unless you managed to get your hands on Dell's recent model, many contain a myriad of shortcomings that have often caused users a lot of frustration and annoyance, rather than enjoyment. However, the latest pre-loaded Linux laptop to make its debut could just be the game-changer we've been waiting for.

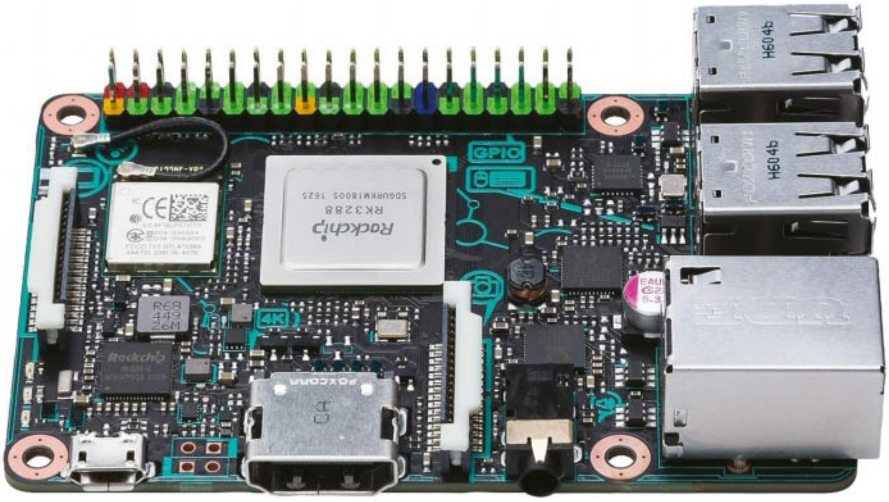
Ever-popular KDE, famed for its fantastic graphical desktop environment, is launching its very own laptop. To do this, it's partnering with the Spain-based Slimbook for the official launch of the KDE Slimbook. In the past, KDE has been the desktop environment of choice for a vast majority of UNIX workstations, providing one of the most accessible ways to get started with everything that Linux offers. Slimbook, on the other hand, won't be overly known to those outside of Spain, but it has previously built and sold entry-level laptops with varying levels of success.

> " The latest pre-loaded Linux laptop could just be the game-changer we've been waiting for "

The word budget won't apply to the KDE Slimbook, however, as it's being marketed as a mid to high-level desktop PC replacement. Early models allow for two variants to be sold, with a choice of a 2.3GB Intel Core i5 or 2.5GB i7 model available for users. Other noticeable specifications include 4GB of RAM, a 13.3-inch 1080p display, 120GB SSD and dual USB 3.0 ports. On paper, these are pleasing specifications, and at just 0.71 inches thick, it's going to rival the same shape and design of leading laptops currently on the market.

While Slimbook is taking care of the hardware side of things, KDE is in its element on the software side. The new KDE Slimbook ships with the popular KDE Neon distribution. While it'll be tailored for new users, thanks to its Ubuntu base, there'll be plenty of scope for advanced users to customise the system to their exact needs. Perhaps the best thing about using KDE Neon on the Slimbook is having access to the KDE Plasma desktop environment, which has proven to be a big success over the years. It will contain a wide range of KDE-based apps, and it won't skimp on the usual personalisation tools associated with the desktop environment. Although details on the matter are still scarce, it's believed that KDE will provide all the necessary tools to switch environments if KDE Neon simply doesn't suit your needs.

What won't be as pleasing to the ears of potential users will be the pricetag. Prices for the KDE Slimbook will be starting at around £700/$770 for the i5 model, with it then increasing closer to the £820/$910 mark for the faster i7 version. Pre-orders are currently being taken over at **kde.slimbook.es**, with the first units expected to ship towards the end of March, but if pre-orders take off, expect some delays.



DEVELOPMENT

Asus' Tinker Board is the Pi competitor we've always wanted

Can 4K capabilities make the Tinker Board the next must-have SBC?

Numerous Raspberry Pi competitors have come and gone over the years, but few have managed to really stand the test of time against the mini computing powerhouse. But despite numerous failures from close competitors, tech giant Asus has waded in with its own Pi competitor, the Tinker Board.

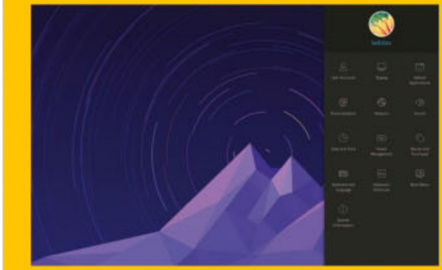
Sizing up at just 8.5cm by 5.3cm, both single-board computers are near identical in size, but that's where the similarities really end. The Tinker Board features a quad-core ARM Cortex A17 CPU at its core, running at 1.8GHz. Alongside this is an ARM Mali GPU and 2GB of DDR3 memory. The Mali GPU will be of particular interest to those who can't decide between this and the Broadcom VideoCore IV GPU of the Pi, with the former providing vastly superior performance in tests. The increase in power aids the Tinker Board to support not only full 1080p, but also H.265 4K decoding, a first for a single board computer of its size. However, in an official statement from Asus, 4K support is still being worked on currently, and while users can technically enjoy 4K video through the unit, don't expect to be watching 4K Netflix anytime soon.

Aside from the core components, other niceties include a couple of swappable antennae for the primary Wi-Fi module that's also on board, along with gigabit Ethernet and key support for SDIO 3.0. Alongside the release of the Tinker Board, Asus has also released its own OS to accompany it. At the time of writing, details of the OS are fairly scarce, however it's known that it's based on Debian and will have a close resemblance to the core Raspberry Pi OS many of us have become accustomed to. Asus has also noted that it's working on support for OpenSUSE and Ubuntu, with expansions planned towards the end of 2017.

The Asus Tinker Board is a tempting piece of kit on paper, and looks to not only match the Raspberry Pi in core features, but also surpass it in other areas. Interested parties can currently pre-order the unit directly through Asus for the tasty price of just £55, with early units shipping now.

TOP FIVE

Best Debian-based distros



1 deepin

Arguably deepin's biggest selling point is its self-developed desktop environment, DDE. While based on the Qt 5 toolkit, DDE can be customised to suit the needs of the user, without proving to be a burden on your hardware. Make sure you also take a look at all the home-grown applications it offers, including the fantastic DMusic and DPlayer apps.

2 Tails

If security is important to you, then Tails should be your first distro of choice. It uses a clever Incognito mode, which helps you bypass internet traffic and makes tracking you only a faint possibility.

3 antiX

While not as well-known as the others on this list, antiX shouldn't be discarded. It's primarily built for older hardware, but still provides a fully functional OS for both newcomers and seasoned users alike.

4 SparkyLinux

SparkyLinux has gained a growing reputation among those seeking a simple distro that doesn't contain a mountain of superfluous extras. It's lacking in some core applications, but a few downloads will sort this issue.

5 Parrot Security OS

Another one for the security-minded folks out there, Parrot includes a myriad of utilities built for penetration testing, computer forensics and even reverse engineering. This isn't one for those new to Linux, however.

RASPBERRY PI

Google is bringing AI to the Pi

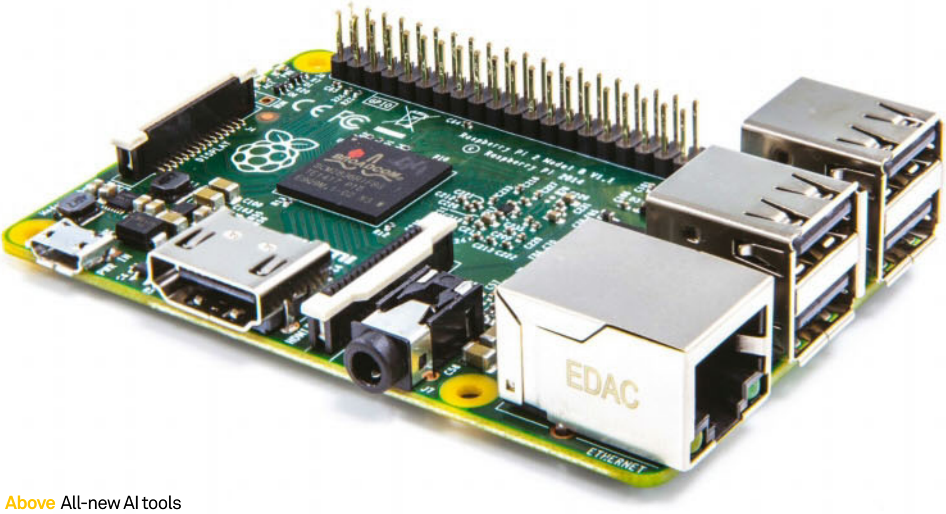
Experimenting with artificial intelligence is set to become a whole lot easier

A bridge between Google and the Raspberry Pi Foundation is something fans have wanted for a while now, and recent developments have shown that the latest news to come from Google HQ may be the closest we've got to this partnership yet.

The tech giant is looking to invest heavily in expanding on its existing suite of development tools that are currently available to Raspberry Pi developers. In turn, this will help better implementation of the Pi in Google-based projects, offering a range of new software tools for natural language processing, predictive analytics and sentiment analysis.

To help scope out the landscape, Google is currently running a survey over on the Raspberry Pi Foundation's website where users can leave their thoughts and suggestions on the idea. "We at Google are interested in creating smart tools for Makers, and want to hear from you about what would be most helpful," it says at the beginning of the survey. As well as all of the areas listed previously, the survey also outlines the other areas where Google is keen to work with the Pi. Areas like 3D printing, IoT development and drones have all been explored with the Pi, but with Google's AI tools on board, who knows what users could possibly achieve?

> “We at Google are interested in creating smart tools for Makers, and want to hear from you”



Above All-new AI tools could soon be a feature of the Pi

This partnership is big news, and it's seen as a great way to accelerate the development in this field in the near future. It has previously invested a lot of time in the Pi, with Google gifting $1 million worth of microprocessors to UK schools back in 2013. Budding developers should head across to the Raspberry Pi Foundation site for more details.

SOFTWARE

Getting Windows on Linux has never been easier

Finding an accessible way to successfully run Windows apps on Linux was for a long time a pretty thankless task. WINE, however, looked to turn the whole process on its head. It offered users a relatively easy way to run the Windows API on top of a Linux-based OS. Early versions were ropey, with bugs galore and questionable usability at times. It also suffered from limited availability outside of Debian and Ubuntu-based distributions, but that has also thankfully changed.

Thanks to a wave of interest in the community, the latest version of WINE, 2.0 to be exact, is now available for download. At the core of the latest update are improvements to the stability of running thousands of Windows programs on your Linux desktop. There's more than 60 key bug fixes, and improvements have been made to overall compatibility with some of the more obscure distros around.

Arguably the biggest update is the fact that Microsoft Office 2013 now has full WINE compatibility, providing an unrivalled document-creation service. It's however, only one of the many new apps that up until now, would be out of reach for most Linux users. Early user feedback has shown there are still some issues with larger pieces of software, but that overall usability is vastly superior compared to earlier builds of the program.

Initial binary packages for the likes of Debian, Mageia, Fedora and Ubuntu are all available for download, with further expansion planned in the next development cycle. The official WINE 2.0 source code can be downloaded over at **www.winehq.com**

> “Microsoft Office 2013 now has full WINE compatibility, providing an unrivalled document-creation service”

DISTRIBUTION

Linux Mint 18.1 KDE and Xfce editions now available

Latest update follows popular MATE release

Towards the end of 2016, the Linux Mint Project unveiled the Cinnamon and MATE versions of the amazing Linux Mint 18.1 OS. The public response has been fantastic, especially due to the long term support plans put in place by the development team, with work on the 18.1 update expected to continue well into 2021.

While the unveiling of the Xfce and KDE editions won't be a surprise, the changes involved are great to see. The Xfce version in particular is based on the latest 4.12 desktop environment, equipped with a wide range of XApps, including the likes of Xplayer, Xreader and Xed to name just a few.

On the other side, the KDE version uses the Plasma 5.8.5 desktop environment, and includes a wide range of improvements to various menus and core settings. Both updates have gone down a storm so far, with more users than ever adding to Mint's growing reputation. Plans are in place to provide further updates to these editions down the line, but as of yet, no concrete details have been released.

Parties interested in either the Linux Mint 18.1 Xfce or KDE editions can head across to the official Mint website (**www.linuxmint.com**) for all necessary download links and instructions for installation for users moving to Mint for the first time.

KERNEL

Linus Torvalds announces latest 4.10 kernel release candidate

Regular updates are coming thick and fast from the king of Linux

We're being spoilt with the amount of release candidates coming from Linus Torvalds in recent weeks. The latest announcement coming from Linux HQ shows off the new release candidate in the 4.10 kernel series.

In an official post, Linus Torvalds has said: "Things seem to be calming down a bit, and everything looks nominal. There's only been about 250 changes (not counting merges) in the last week, and the diffstat touches less than 300 files (with drivers and architecture updates being the bulk, but there's tooling and networking too)."

While most of the latest release candidate is a refinement of current driver improvements, more surprising is the disclosed information about future updates. Included in the official announcement, Torvalds has also revealed plans for a regular release schedule for Linux 4.10, with a total of seven release candidate builds in the pipeline.

By our estimates, this could see the full, final release of Linux 4.10 land towards the end of March. Of course, these plans could soon change, but we'll let you know if they do.

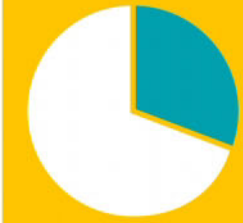
DISTRO FEED

▶ Top 10

(Average hits per day, 15 Jan– 15 Feb)

1.	Linux Mint	▲	3,102
2.	Debian	▲	2,094
3.	Manjaro	▼	1,899
4.	Ubuntu	▲	1,489
5.	openSUSE	▲	1,256
6.	Solus	▼	1,053
7.	Zorin	▲	1,005
8.	Fedora	▲	998
9.	Antergos	▼	927
10.	elementary	▲	917

▶ This month



Downloads are generally up this month, with Mint extending its lead at the top of the distro food chain. It's also great to see some emerging distros make their way into the top ten.

▶ Highlights

Linux Mint
Mint is growing at a rapid pace, so much so that it's hard to keep up with it! Its recent 18.1 update has been widely praised by the community, primarily for the amount of changes that have been introduced. If you haven't checked it out already, make sure you do soon.

Solus
Solus keeps climbing up the board of top downloads, and it's easy to see why. Touted as being one of the best distros for low-resource machines, Solus prides itself on its ease-of-use and beautiful design.

elementaryOS
We can't talk about beautiful distributions without also mentioning elementaryOS. Its desktop environment, titled Pantheon, is arguably one of the most user friendly environments we've ever used.

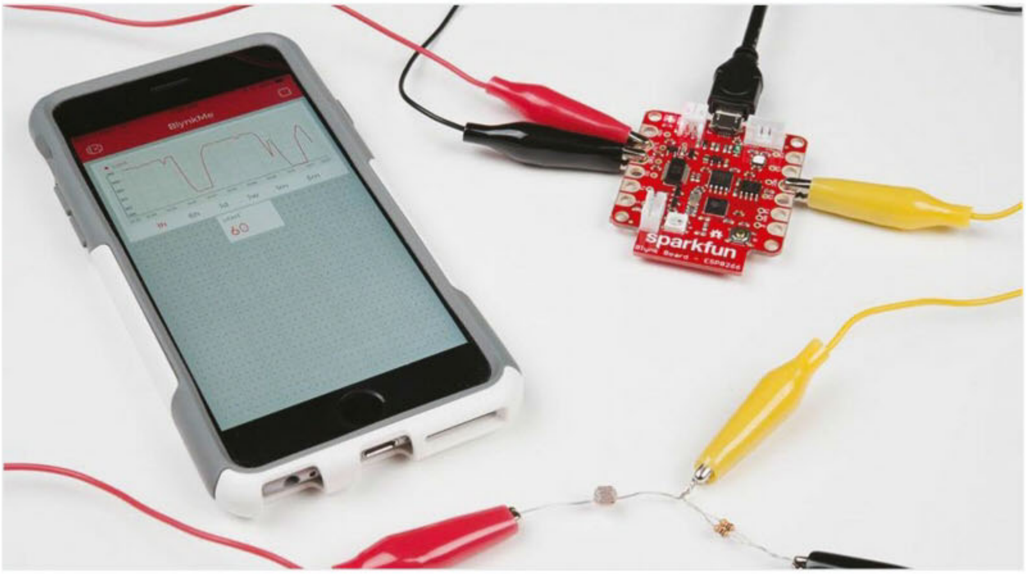
Latest distros available:
filesilo.co.uk

INTERVIEW PAVEL BAIBORODIN

Build an app in ten minutes

This new platform aims to give you all the tools you need to build an app for your Arduino project in ten minutes or less. Pavel Baiborodin gives us the lowdown on why Blynk is a game changer for budding developers

Pavel Baiborodin
has a history in development with microcomputers and experience with UX design. His desire to create Blynk comes from wanting to take the complexity out of developing for both Arduino and the Raspberry Pi. A get started guide is available on the website at **www.blynk.cc**



Where did the original idea for Blynk first stem from?

I first started tinkering with microcomputers four years ago and I was blown away with how quickly you can start prototyping physical interactive objects these days. Having a background in user experience and design, I quickly realised that I needed some sort of a visual interface to control my projects, but I was frustrated to discover that there is no easy way to do that. So, you just got your first LED blinking (by making baby steps in coding) and then you realise that you need to learn PHP, Xcode, Android and all the other things. Not to mention that most of the tools in this space were made by developers and for developers.

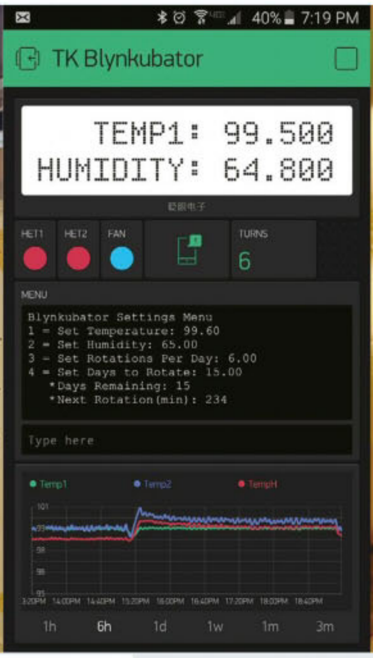
This is where I decided to fix that. I told Dmitry, my co-founder and CTO, about the idea, and we started building it for ourselves. The first project ever made was a watering system to keep my plants alive while I'm travelling. Of course it could have been automated, but I just enjoyed pressing the button to start the pump [while] in another part of the world. Then we had a Kickstarter campaign, grew to [a community of] more than 200,000, and got billions of requests on our servers every month.

For those who may not have heard about Blynk, could you give us an overview of its key features?

Blynk is the easiest and most adopted platform to build mobile applications for IoT projects and connected products. There are three major components: firmware, server and mobile apps. First of all, we provide libraries that can get almost any hardware online, (currently more than 400 models) no matter what connectivity you choose. Wi-Fi, cellular, Ethernet and XBee are all already online, and for local connection you can use BLE [and] USB. Plus, I've [forgotten to] mention the likes of Node.js, Python and C.

Blynk also supports industrial communication protocols like RS232, RS485, UART, CAN, ModBus [and] OneWire. When your hardware is online, it would connect to our Blynk Cloud Server, which is open source and was built on Java. Access to Blynk Cloud Server is free, however you can also run it locally. In essence, the server connects your hardware and smartphone together.

The local Blynk server can even run on a $5 Raspberry Pi, where it will handle up to 1,000 requests per second. It's secure, lightweight and pretty damn fast. Then comes the mobile apps for iOS and Android where one can quickly snap together an interface to control things



TK Blynkubator

```
TEMP1: 99.500
HUMIDITY: 64.800
```

```
Blynkubator Settings Menu
1 = Set Temperature: 99.60
2 = Set Humidity: 65.00
3 - Set Rotations Per Day: 6.00
4 = Set Days to Rotate: 15.00
   *Days Remaining: 15
   *Next Rotation(min): 234
```

remotely and visualise sensor data. We've made it super-easy to use: you just drag and drop widgets like buttons, sliders, gauges, charts, maps and many others to create the interface you need. No coding or design skills are required as all the major UI configurations are made on the app side.

Blynk is packed with features designed specifically for the IOT, such as controlling multiple devices and groups, easy sharing access to hardware with other people, syncing physical and HMI states, real-time updates of UI on all the connected clients, push notifications, emails and other great stuff. Blynk does a lot, as you can probably tell.

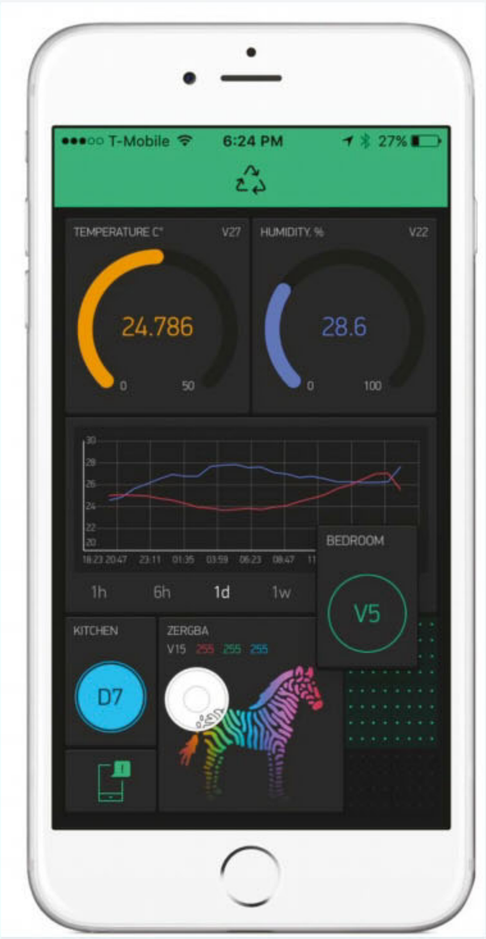
Developing a product like this seems like it's remarkably complex. Was it difficult getting it from concept to reality?

It was, and still is incredibly challenging, but on the other hand, it is remarkably interesting and exciting. From the very beginning we set a really high bar for every component of the system. The first challenge I would mention is maintaining the same level of simplicity and flexibility in UX when such a complex product like Blynk continues to evolve every single week.

Secondly, there are thousands of devices with their own specifics and protocols, and being completely hardware and connectivity agnostic means a lot of work on the firmware. Volodymyr, who is our embedded solutions architect, has developed a solid base for it and now it's pretty easy to add new hardware, but there are always small details we need to keep track of. Even a few of our competitors are using it.

In general, 'connecting the dots' and providing a seamless developer experience is the biggest challenge.

> “ Blynk is the easiest and most adopted platform to build mobile applications for IoT projects and connected products ”



Are there any requirements (hardware or software-based) that users need to be wary of before starting their projects?

Blynk can work on almost any microcontroller or microcomputer. From Arduinos and Raspberry Pis to Particle Photon, Intel Edison and the like. You can check the list of supported hardware on our website (**www.blynk.cc**) for starters. You would also need some sort of connectivity that would enable the connection to the internet, which you'd be surprised how many people forget. Luckily, today there are all sorts of devices with connectivity on board. In case you don't have one, BLE is also supported. And if you want to go wired, USB is your choice, however it's a bit tricky for newbies.

If you would like to run your own Blynk server, the only requirement for the hardware is to install Java 8+. For Linux we recommend using our Node.js module.

Blynk apps are running on all iPhones with iOS 8 and above and the Android app requires OS 4.0.3 and above. We don't have a tablet version yet, however you can still launch Blynk on iPads and Android Tablets – it will be just be zoomed in.

Does Blynk offer compatibility for different types of Linux-based hardware?

As for Linux compatibility, we support single board computers like Raspberry Pi, BeagleBone, CHIP, Omega, and Intel boards out-of-the-box. You can also run Blynk on regular distributions like Fedora and Ubuntu, and we hope to expand this library in the near future.

The Node.js module for Linux installations would be our recommendation to use, because it proves to be way easier to implement simple projects using JavaScript, however, [a] pure C++ library is available as well.

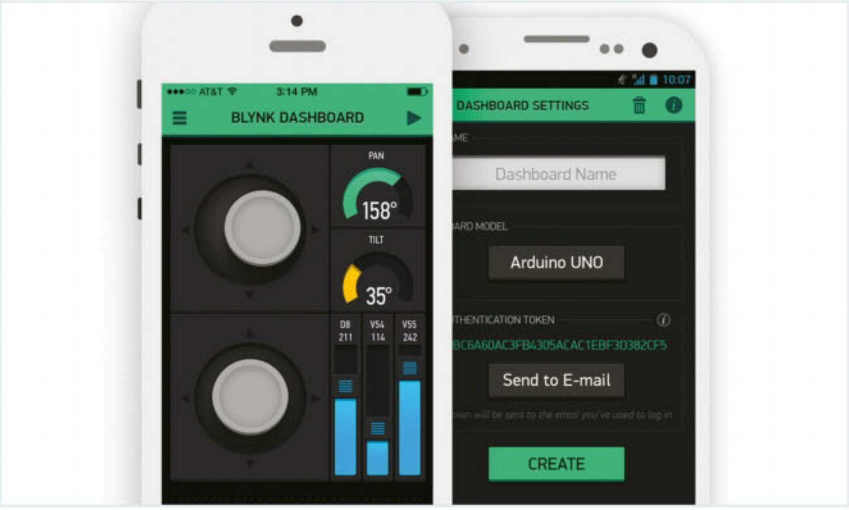
I would also like to mention that Blynk provides an installation package for OpenWRT, expanding to hundreds of Linux-based network devices.

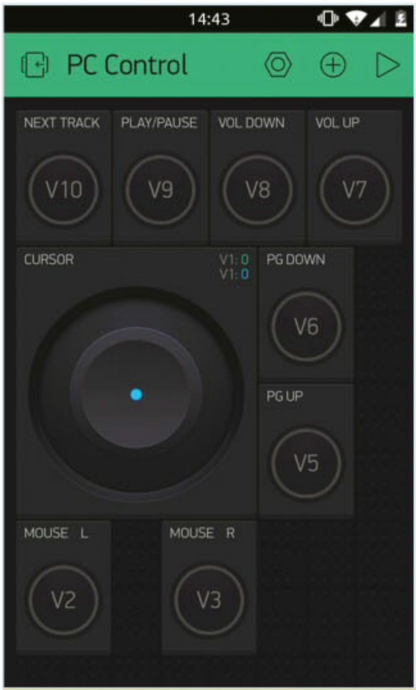
Is Blynk suitable for those coming across to Arduino development for the first time?

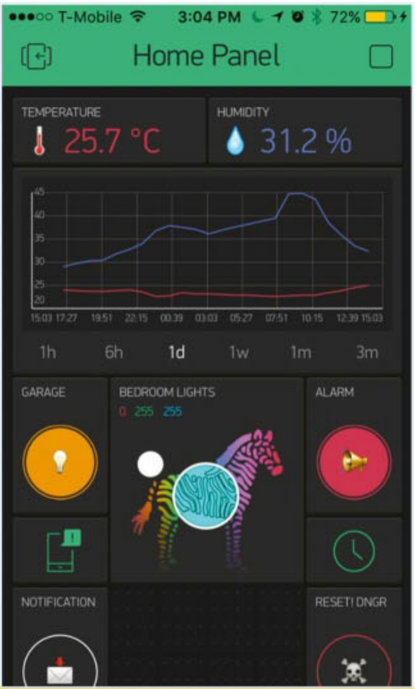
When we started Blynk, Arduino and Raspberry Pi were the only two target platforms, and we wanted our users to be able to get first results in under five minutes. I think we did a great job with a really simple start. All that [is] needed is to paste the special key (we call it Auth Token) to the example code we provide, and upload the firmware to the device. After that you instantly get access to sensor data from analog inputs, [and you] can control digital pins and send PWM signals straight from the app. No coding is involved at all.

As you explore more features, you might need to refer to documentation or the tons of tutorials on Hackster, Instructables, YouTube and so on. Our community page is full of smart and positive folks who are always ready to help as well.

We saw some mention of the Virtual Pins system Blynk can use, could you tell us a little more about this?

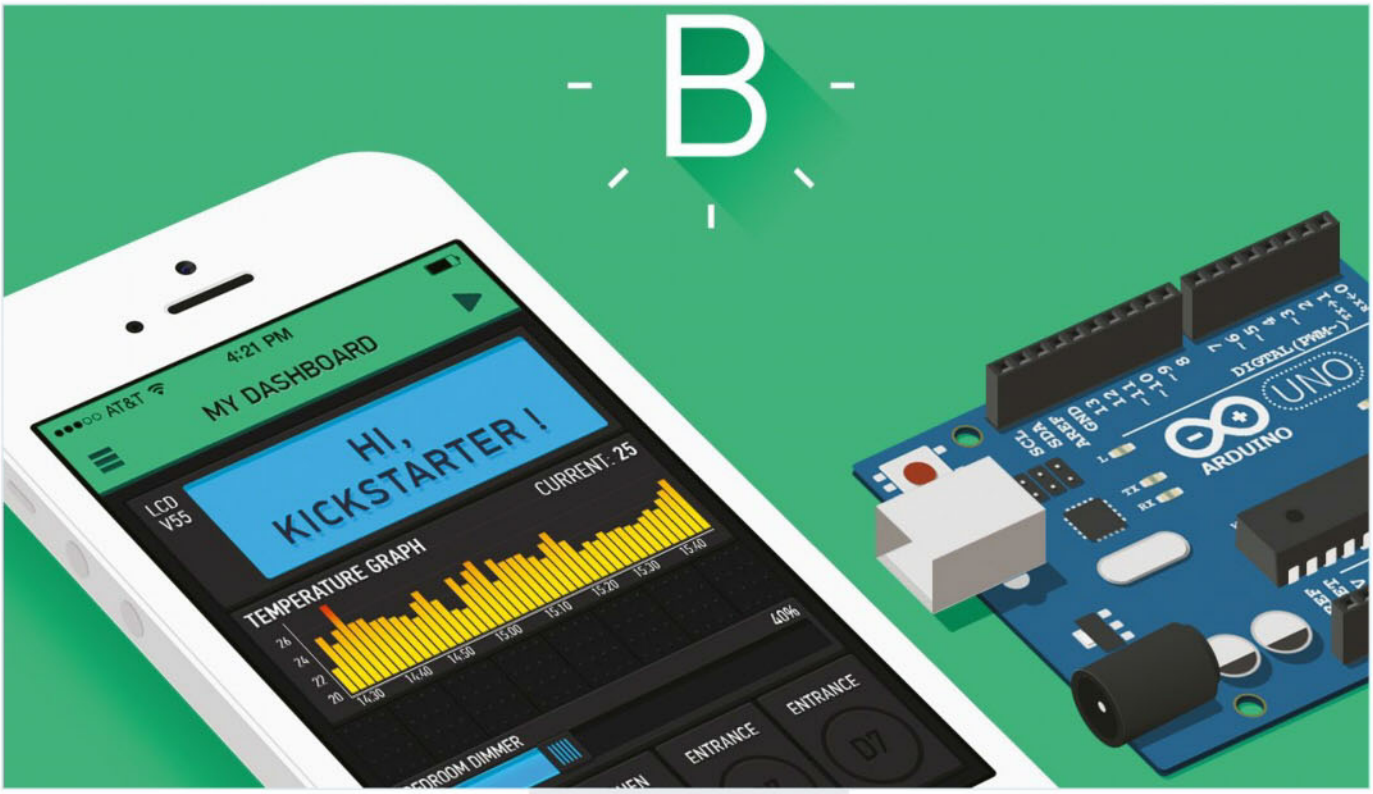






The wonderful world of widgets

The core of most projects created through Blynk will be primarily made of widgets. Many of the widgets already designed for Blynk offer triggers for certain features, with others connecting and syncing to your smartphone. Perhaps what makes the choice of widgets in Blynk so intriguing is that the team has also introduced a variety of sensors. So if you find your project requires the use of GPS, an accelerometer, or perhaps a gyroscope, there's a pre-programmed widget to get it up and running. The team is planning to bring out further widgets in new updates to really start exploring some of the hardware possibilities they've yet to cover. We recommend heading across to the Blynk forums to see some of the unique ways that users have introduced widgets into their projects.



Virtual Pins is a really simple yet incredibly powerful concept we invented. As I already mentioned, the Blynk app can control and read physical GPIOs without any additional code. However, there are cases when data should be processed on the device, or [an] additional library is required to read a specific sensor.

For example, [what] if you need to read an I2C temperature sensor, convert the readings from Celsius to Fahrenheit and calculate the average temp per minute? First, this sensor is not attached to a specific analog input pin, secondly you might need a library that would read this particular sensor model, and then you would need to perform an averaging operation on board. This is where Virtual Pins come into play. They act as a channel to send any data from the hardware to the apps and vice versa. On the hardware side, you would push the processed temperature to, let's say Virtual Pin 1. On the app side, you would add a Graph Widget and configure it to use the same Virtual Pin 1 and plot the incoming data easily.

Or, imagine you need a single button in the app that will automate the process of dimming the lights, closing the window blinds, and turning on your TV. This button would send a command to Virtual Pin 2, and on the hardware side this command will trigger a function that will perform all the actions one by one. It gives a lot of flexibility to developers, because no matter what is connected to the hardware, it can be controlled or read from within the app.

> “Blynk is a very dynamic product. Our roadmap for 2017 is already packed with new features and enhancements ”

What are some of the best ways you've seen Blynk put into action already?

There are thousands of projects made with Blynk and products that utilise [the] Blynk platform. It's really difficult to choose the best one. They vary from connected homes, chemical dispensers, boilers, connected hatcheries, home beer breweries, drones and many others. It's really exciting to wake up in the morning and see another great idea put into reality with Blynk. This is what keeps us inspired every day.

Do you have any plans for Blynk in the future? Perhaps some areas you'd like to expand on?

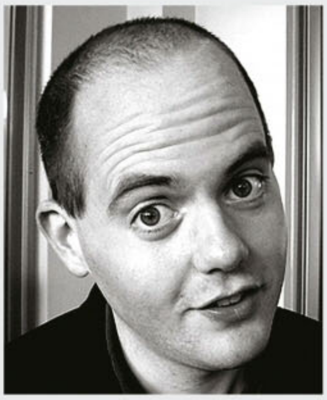
Blynk is a very dynamic product. Our roadmap for 2017 is already packed with new features and enhancements. We carefully listen to our community and many ideas are generated by them. There will be more customisation features, new widgets, UI enhancements and other cool stuff.

We are also working on a really exciting service: a publishing platform for apps made with Blynk. Anyone will be able to export their Blynk project into a standalone app and then publish it to the App Store and Google Play in just a few days! We believe that it will explode the creation of new connected products made by talented developers and engineers, and it will boost the IoT growth further. So stay tuned, as this year is set to be exciting. ■

OPINION

The kernel column

Jon Masters summarises the latest happenings in the Linux kernel community



Jon Masters
is a Linux-kernel hacker who has been working on Linux for some 19 years, since he first attended university at the age of 13. Jon lives in Cambridge, Massachusetts, and works for a large enterprise Linux vendor, where he is driving the creation of standards for energy efficient ARM-powered servers

Linus Torvalds announced a rare eighth release candidate of Linux 4.10. This wasn't due to massive instability in 4.10, but rather as a consequence of timing. Many of us (including Linus) in the Linux community were, at the time, in Tahoe, California for the annual Linux Foundation Open Source Leadership Summit (OSLS, formerly known as Collab Summit). Had the 4.10 final been released that week, it would have immediately opened the merge window for many thousands of patches destined to land in 4.11. Although he has juggled travel with managing a merge window before, Linus "prefers not to", and who are we to blame him?

It is worth noting that a couple of minor bugs have been reported against 4.10-rc8, including one nfsd warning splat that Dave Jones reported (and was immediately fixed), and one boot time intermittent hang that Pavel Machek flagged on his primary x86 system. He is still working on 'bisecting' the problem by continually building test kernels that bisect through the latest changes to the kernel since a previously known, good version. Bisection can be very laborious, but it will usually find the problem, provided that it can be reproduced fairly reliably in every iteration. Neither of these problems would generally hold up the 4.10 final, however.

Coherent device memory nodes

The computing industry is currently undergoing a major technology inflexion as the rate of acceleration of compute performance across generations slows; yet customer and user demand for computation has never been higher. As a direct result of this dichotomy, active research in compute accelerators is gaining momentum. Workload accelerators (as they are sometimes called) are traditionally discrete pieces of silicon that provide some offload capability for performing functions that would otherwise waste CPU cycles. A classic example is that of an encryption or a compression engine, such as those contained within many contemporary network cards. The cards are said to offload certain calculations that might otherwise burden the CPU.

For the past few years, workload acceleration meant widgets added to certain chips (whether server examples like this example, or power efficient capabilities added to mobile chips to save energy – such as media playback/recording codecs implemented in hardware), or (perhaps more commonly) the rising use of GPUs in a GPGPU (General Purpose) configuration. Nvidia's Tesla and AMD's FirePro come to mind, but there are others. In recent times, however, the range of options for acceleration has begun to balloon, especially with the nascent adoption of once exotic technologies, such as FPGAs.

An FPGA (Field Programmable Gate Array) is non-fixed function silicon containing millions (or even more) of uncommitted gates that can be dynamically wired up in almost any configuration. This allows for custom hardware logic to be implemented rapidly without actually going to the expense of designing and manufacturing a full chip. Of course, there's a downside to using FPGAs (or nobody would bother with making chips of their own any more): they use more energy than full custom chips, are slower than a custom chip, and they cost more to produce and to purchase. Nevertheless, FPGAs can implement logic that executes far faster than contemporary CPUs, and even beats out GPUs in some cases. Couple that with an ability to dynamically reconfigure them without touching the underlying hardware, and FPGAs become very interesting indeed. They represent just one of a whole host of novel approaches that are gaining traction to offset the relative lacklustre growth in CPU performance.

These novel devices – whether (GP)GPUs, FPGAs, custom silicon (for example, inGoogle's TPUs – Tensor Processing Units) need to be attached to the rest of the system. In the past, this happened through an interconnect such as PCIe, through which data could be exchanged (using a built-in DMA – Direct Memory Access – engine) under the control of a device-specific driver. While this works, it can be highly inefficient once large amounts of memory exist on a device (in theory, even more memory than is installed as system memory on the host/PC side of the link). A number of emerging interconnects aim to improve performance by bringing memory coherency on both sides into the mix, allowing for devices to directly share data with applications without first copying it around.

Coherency means that a single coherent view of memory is seen by both a device and the rest of the system. Memory that is installed on a specific adaptor card is accessible to the host just as the host memory is accessible to the device (albeit perhaps with security restrictions) and changes to that memory are immediately visible on either side of the interconnect. Implementing coherency requires a coherency protocol and a large amount of complexity, but the upshot is that device memory can be managed just like host system memory. Which brings us to an RFC (Request For Comments) patch series posted by Anshuman Khandual of IBM. His 'Define coherent device memory node' aims to extend the existing kernel notion of NUMA (Non-Uniform Memory Access) to support the creation of device nodes that contain struct page structures managed by the Linux kernel in much the same way that it manages normal RAM today.

The use of NUMA is interesting. In a NUMA machine (think multi-socket server), all parts of the system share a global memory, but may experience relative differences in performance for accesses that aren't local to the NUMA node. Thus a CDA (Coherent Device Node) in the new world order would be able to see the same memory as another node containing CPU(s), but Linux would know that such accesses may go across an interconnect. Anshuman's patches are very interesting because they start to implement a reality many of us have wanted for years: allowing applications software to directly map accelerator memory without the need for special device drivers at all. It's early days, but this kind of work could eventually make using accelerators much easier. The full thread is available here: **https://lkml.org/lkml/2017/1/29/198**.

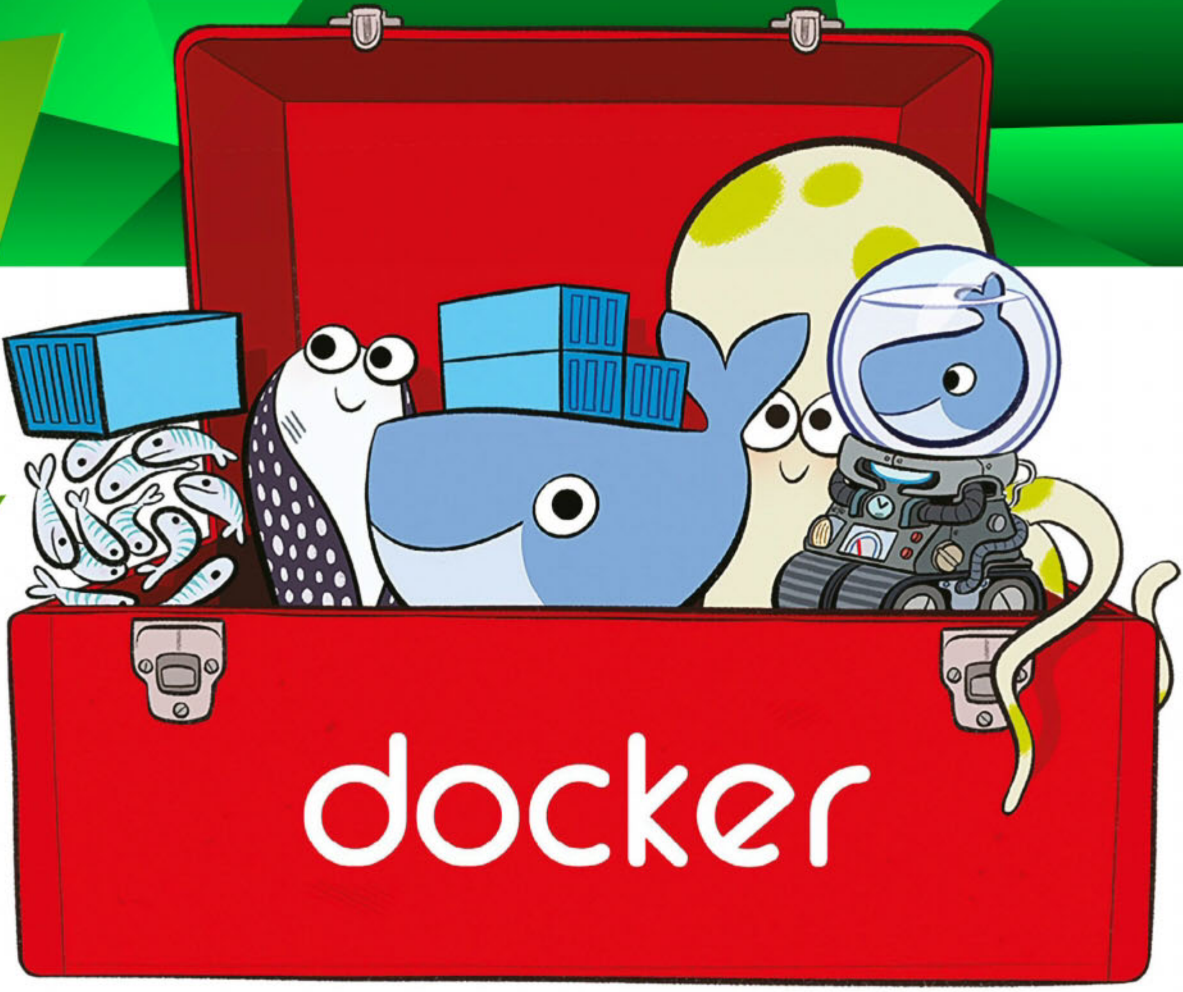
Ongoing development

Kernel development rarely stands still (perhaps it grinds to a crawl during the annual Kernel Summit for those who are attending) and this month was no different. Two discussion threads caught your author's attention. One had to do with the implementation of error handling for VFIO, the other focused on ongoing work to better understand and to document memory ordering in Linux.

VFIO (Virtual Function IO) is a Linux kernel abstraction that generically represents IOMMUs (IO Memory Management Units) and allows virtual functions provided by adaptor cards (such as PCIe) to be assigned into guest virtual machines. Such VMs see the performance of dedicated hardware while retaining the management and isolation benefits afforded through virtualisation. In the latest round of work, Michael S Tsirkin seeks to extend the existing limited support for handling PCIe AER errors into a generic mechanism to report underlying host link errors into VMs.

Meanwhile, Paul McKenney (author and co-inventor of the RCU Read Copy Update mechanism used within the Linux kernel for lock-free scalable data structures, as well as a wonderful free online book on parallel programming known as the 'perf book') has continued his work to describe the memory ordering models used by various computer architectures. Most people are completely unaware of memory ordering and how it works because they are used to the strong model employed by Intel x86. Many alternative models exist, however, and they have implications for low-level primitives (such as RCU) in terms of how changes to memory are observed between multiple processors (agents, processing elements, or observers), and in what order. Paul is working with a number of others (such as Will Deacon of ARM) to better describe these models and capture their semantics. Most recently, he was fishing for access to a live DEC Alpha machine to test out some assumptions.

Finally this month, **kernel.org** (the site that hosts much of the Linux kernel development, and is supported through the donations of companies and others who want to support Linux) announced that it is finally discontinuing FTP service this year. Beginning 1 March, 2017 it will no longer serve out kernels via FTP. Mirrors.kernel.org will follow in December. FTP isn't the only service from a bygone era that kernel.org has implemented over the years. It once implemented a public NFS/CIFS (yes, Windows) server, but that was a fairly brief occurrence for obvious (security) reasons. ■



TAKE CONTROL OF CONTAINERS

Containers and container orchestration platforms are revolutionising DevOps and infrastructure. Find out more about how you can take advantage of them

Containers offer a solution to the problem of software portability, i.e. how to get it to run reliably when moving from one computing environment to another. Software development is a lengthy process and needs several iterations of testing and development before it can be termed production-ready. This will usually involve working in numerous different environments – from a developer's PC to a test environment, from staging to production, and from a physical machine in a data centre to a virtual machine in the cloud.

> “A container holds an application and everything needed to run it, bundled into a single package”

A container holds an application, along with all its dependencies, libraries and other binaries, plus the configuration files needed to run it, bundled into a single package. By containerising the application and its dependencies, differences in operating systems and infrastructure are abstracted away. Anyone can run the container image from their system without worrying too much about the underlying requirements.

As development and production practices move towards containerisation, it is important to understand that while containers are lightweight and great for porting applications, it is difficult to manage containers in a production environment, which may well have thousands or more containers with different application components running. This brings container orchestration tools into the picture. These are systems to manage all of the bare-metal or virtual machines that you need to run your containers on. They also manage your containers, launch them on the underlying machines, make sure they are distributed and keep them healthy.

CONTAINERS VS VIRTUAL MACHINES

CONTAINERS

- House the minimal requirements for running an application, enabling quick and lightweight deployment.
- Independent self-sufficient application bundles; they can be run across machines without compatibility issues.
- Can be versioned, archived, shared and used for rolling back previous versions of an application. Platform configurations can essentially be managed as code.
- Help break down deployment into functional discrete parts, making a clear separation of concerns (but this also means more components to handle).
- More containers can be put onto a server than onto a traditional virtual machine.
- Many containers running on one host system can in theory cause security concerns, as they have access to the same kernel.
- A resource-light way to get started with a program.

VIRTUAL MACHINES

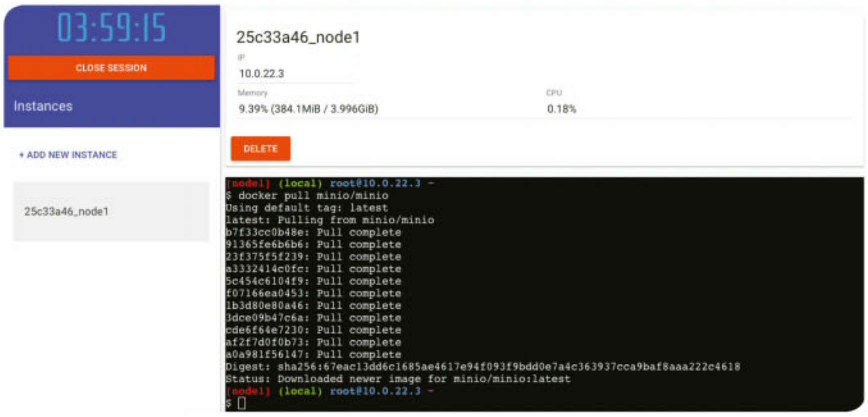
- Run many discrete OS instances in parallel on a single host with VMware.
- Emulate virtual hardware and account for all the underlying system requirements.
- VM images are significantly larger than containers.
- Enable true hardware-level isolation, so the chance for interference and/or exploitation is theoretically less likely compared to containers.
- Ensure security and consistent OS interface.
- In real-world scenarios, virtual machines and containers are complementary and there is a high chance that you'll need a combination of both.

CONTAINERISE A COMPLETE FILE SYSTEM WITH **DOCKER**

Docker is the most renowned name in container technology. Here's why…

Docker containers wrap a piece of software in a complete file system that contains everything needed to run it: code, runtime, system tools, system libraries – anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment. One of the USPs of Docker containers is that they start instantly and use less RAM. Images are constructed from layered file systems and share common files, making disk usage and image downloads much more efficient.

Docker uses the Linux kernel and its features, like cgroups and namespaces, to segregate processes so they can run independently. This independence is the intention of containers – the ability to run multiple processes and apps separately from one another to make better use of your infrastructure while retaining the security you would have with separate systems. Docker technology was initially built on top of LXC, generally thought of as traditional Linux containerisation. LXC was useful as lightweight virtualisation, but it didn't have a great developer or user experience. Now, Docker technology brings more than the ability to run containers – it also eases the process of creating and building containers, shipping images and versioning images.



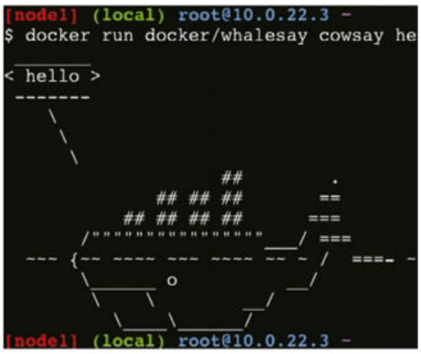
25c33a46_node1

IP
10.0.22.3

Memory
9.39% (384.1MiB / 3.996GiB)

CPU
0.18%

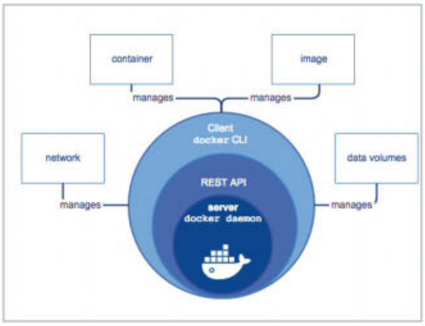
```
[node1] (local) root@10.0.22.3 ~
$ docker pull minio/minio
Using default tag: latest
latest: Pulling from minio/minio
b7f33cc0b48e: Pull complete
91365fe6b6b6: Pull complete
23f375f5f239: Pull complete
a3332414c0fc: Pull complete
5c454c6104f9: Pull complete
f07166ea0453: Pull complete
1b3d80e80a46: Pull complete
3dce09b47c6a: Pull complete
cde6f64e7230: Pull complete
af2f7d0f0b73: Pull complete
a0a981f56147: Pull complete
Digest: sha256:67eac13dd6c1685ae4617e94f093f9bdd0e7a4c363937cca9baf8aaa222c4618
Status: Downloaded newer image for minio/minio:latest
[node1] (local) root@10.0.22.3 ~
$ 
```



```
[node1] (local) root@10.0.22.3 ~
$ docker run docker/whalesay cowsay he
 _______
< hello >
 -------
    \
     \
      \
                    ##         .
              ## ## ##        ==
           ## ## ## ##       ===
       /""""""""""""""""""___/ ===
  ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~
       \______ o          __/
        \    \         __/
          \____\_______/
[node1] (local) root@10.0.22.3 ~
```

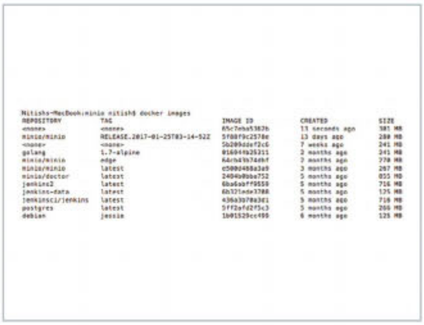
01 Using Docker

Docker is used for fast and consistent delivery of your application. Docker supports continuous integration and a continuous deployment workflow. When the testing is complete, to deliver the product you can just push the updated image to a production environment. Being a container-based platform, Docker allows highly portable workloads. As Docker is lightweight and portable, it is easy to dynamically manage workloads and scale up or tear down applications and services as businesses require.



02 Docker architecture

Docker uses client-server architecture. The Docker daemon does the heavy lifting of building, running and distributing your Docker containers. The Docker client communicates with the Docker daemon using REST API over UNIX sockets or a network interface. The Docker client has a command-line interface which it uses to talk to the daemon. Docker contains images, containers and registries in the daemon process. A Docker image is a read-only template that has instructions for creating a Docker container. A Docker container is a runnable instance of a Docker image, while a Docker registry is a library of images.



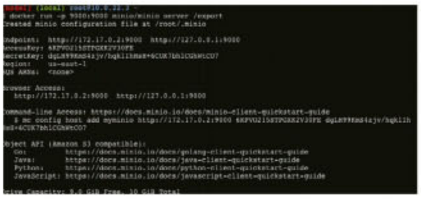
```
Nitishs-MacBook:minio nitish$ docker images
REPOSITORY          TAG                              IMAGE ID            CREATED             SIZE
<none>              <none>                           65c7eba5387b        11 seconds ago      381 MB
minio/minio         RELEASE.2017-01-25T03-14-52Z     5f88f9c2578e        13 days ago         280 MB
<none>              <none>                           5b209ddef2c6        7 weeks ago         241 MB
golang              1.7-alpine                       01604fb25311        2 months ago        241 MB
minio/minio         edge                             64cb47b74dbf        2 months ago        770 MB
minio/minio         latest                           c5000488a3a9        3 months ago        287 MB
minio/doctor        latest                           2404b0bba752        5 months ago        855 MB
jenkins2            latest                           6ba6abff9558        5 months ago        716 MB
jenkins-data        latest                           6b321ede3708        5 months ago        125 MB
jenkinsci/jenkins   latest                           436a3b70a3d1        5 months ago        718 MB
postgres            latest                           5ff2afd2f5c3        5 months ago        266 MB
debian              jessie                           1b01529cc498        6 months ago        125 MB
```

03 Docker images

Docker images are the build components of Docker. Each image consists of a series of layers which are combined into a single image using union file systems. These layers make Docker lightweight. If you update an application, a new layer is then built and replaces only the layer that was updated. Every image starts from a base image. The simple, descriptive instructions are stored in a Dockerfile, which is used to build the base image. Each instruction creates a new layer in the image. The base image is defined using the FROM keyword in the Dockerfile.

04 Docker registries

Docker registries are the distribution components of Docker. They can be private or public and can be on the same server as the daemon process or on a remote server. They store Docker images on it. After images are built, they can be pushed to a public or private registry. Docker Hub is a public Docker registry and has a huge collection of images. It allows you to contribute Docker images of your own. Docker Store, which is now generally available after being in beta, allows you to buy and sell Docker images.



```
 docker run -p 9000:9000 minio/minio server /export
reated minio configuration file at /root/.minio

ndpoint:  http://172.17.0.2:9000  http://127.0.0.1:9000
ccessKey: 6KPVO215STPGXK2V30FE
ecretKey: dgLM99Kmd4zjv/hgkl1hMa8+6CUK7bh1CGhWtC07
egion:    us-east-1
QS ARNs:  <none>

rowser Access:
  http://172.17.0.2:9000  http://127.0.0.1:9000

ommand-line Access: https://docs.minio.io/docs/minio-client-quickstart-guide
  $ mc config host add myminio http://172.17.0.2:9000 6KPVO215STPGXK2V30FE dgLM99Kmd4zjv/hgkl1h
Ma8+6CUK7bh1CGhWtC07

bject API (Amazon S3 compatible):
  Go:         https://docs.minio.io/docs/golang-client-quickstart-guide
  Java:       https://docs.minio.io/docs/java-client-quickstart-guide
  Python:     https://docs.minio.io/docs/python-client-quickstart-guide
  JavaScript: https://docs.minio.io/docs/javascript-client-quickstart-guide

rive Capacity: 9.0 GiB Free, 10 GiB Total
```

05 Docker containers

Docker containers are the run components of Docker. Using Docker API or CLI commands, you can run, start, stop, move or delete a container. Each container is built from an image and behaves like an isolated and secure application platform. They can also access resources running on different hosts or containers. The Docker client instructs the Docker daemon to run a container using the **docker run** command or equivalent API. For example, to run a container using an Ubuntu Docker image and execute the /bin/bash command in interactive mode, execute:

```
$ docker run -i -t ubuntu /bin/bash
```

> “Make better use of your infrastructure while retaining the security you would have with separate systems”

06 Docker services

Docker services are the scalability component of Docker. They enable a swarm of Docker nodes to work together. Docker services allow a defined number of instances of replica tasks. These replica tasks themselves are Docker images. You can specify a number of concurrent replica tasks to run. The swarm manager evenly distributes the load across worker nodes, so it appears to be a single application to the consumer. The **docker service** command is used for creating, updating, inspecting, listing and scaling the services. To create a new service, execute:

```
$ docker service create
```

07 Write your own Dockerfile: 1

Create a new file and save it with the name **Dockerfile**. In this file you can enter the instructions available for use in Dockerfile. We will take the example of a stateful application like Minio. First, specify the base image using the FROM instruction:

```
FROM golang:1.7-alpine
```

If you need to specify any environment variables, there are set instructions available for this, such as ADD, COPY, ENV, EXPOSE, LABEL, USER, WORKDIR, VOLUME, STOPSIGNAL and ONBUILD.

Suppose you need the WORKDIR instruction; in this case, enter:

```
WORKDIR /go/src/app
```

This sets the working directory to **/go/src/app**.

08 Write your own Dockerfile: 2

The RUN instruction executes commands in a new layer on top of the current image and commits the results. The RUN command executes in the shell, which is by default **/bin/sh -c**. To execute more than one command append \ and continue on the next line. In our Minio Dockerfile, the RUN instruction looks like this:

```
RUN \
       apk add --no-cache git && \

       go-wrapper download && \ …
```

The EXPOSE instruction specifies the runtime network ports the container listens to:

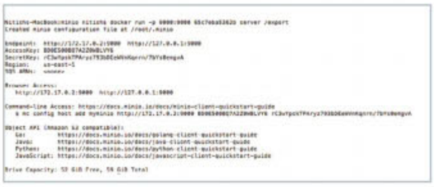
```
EXPOSE 9000
```

ENTRYPOINT configures an executable container:

```
ENTRYPOINT ["minio"]
```

The VOLUME instruction creates a mount point:

```
VOLUME ["/export"]
```



```
Nitishs-MacBook:minio nitish$ docker run -p 9000:9000 65c7eba5362b server /export
Created minio configuration file at /root/.minio

Endpoint:  http://172.17.0.2:9000  http://127.0.0.1:9000
AccessKey: BO0ES0O0B7A22OWBLVYS
SecretKey: rC3wYpskTPAryz7930DEeWVnKgrn/7bYsBengvA
Region:    us-east-1
SQS ARNs:  <none>

Browser Access:
   http://172.17.0.2:9000  http://127.0.0.1:9000

Command-line Access: https://docs.minio.io/docs/minio-client-quickstart-guide
   $ mc config host add myminio http://172.17.0.2:9000 BO0ES0O0B7A22OWBLVYS rC3wYpskTPAryz7930DEeWVnKgrn/7bYsBengvA

Object API (Amazon S3 compatible):
   Go:         https://docs.minio.io/docs/golang-client-quickstart-guide
   Java:       https://docs.minio.io/docs/java-client-quickstart-guide
   Python:     https://docs.minio.io/docs/python-client-quickstart-guide
   JavaScript: https://docs.minio.io/docs/javascript-client-quickstart-guide

Drive Capacity: 52 GiB Free, 58 GiB Total
```

09 Build and run your Dockerfile

Navigate to the folder with your Dockerfile in it. Then use the build command to create the image:

```
$ docker build .
```

This finds the Dockerfile in the current folder and builds it, creating a Docker image. To see all the images, execute:

```
$ docker images
```

This displays repository, tag, image ID, date created and the size details of the image. Now run the image to create a container. For example:

```
$ docker run -p 9000:9000
65c7eba5362b server /export
```

Here, 65c7eba5362b is the image ID of the image that has been built; it is obtained from the Docker images list.

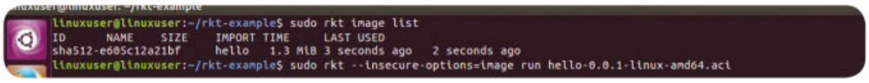
RUN INDIVIDUAL APPS SECURELY WITH **RKT**

rkt is a container offering from CoreOS that can verify container images before running

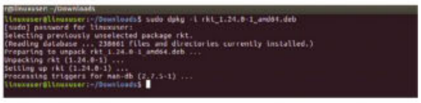
rkt was started by CoreOS as an alternative to Docker. It is an implementation of the app container (appc) specification, and the project was initially called Rocket. rkt was developed as a standalone tool that lives outside CoreOS and is supported on Ubuntu, RHEL, CentOS and other major distros. rkt is composable. It also has crypto primitives for strong trust, image auditing and application identity, making it secure. It can download, cryptographically verify and run application container images. It is not designed to run full system containers, but instead individual applications such as web apps, databases or caches.

rkt enables deployments to private environments without the requirement of a registry. Discovery of container images is simple and facilitated by a federated namespace and distributed retrieval. The format and runtime is well specified and developed by a community. There are independent implementations of tools, which allows the same container to run consistently.

rkt has no centralised daemon to manage containers. It launches containers directly from client commands, making it compatible with init systems such as systemd, upstart and others.



```
linuxuser@linuxuser:~/rkt-example$ sudo rkt image list
ID          NAME    SIZE     IMPORT TIME        LAST USED
sha512-e605c12a21bf     hello   1.3 MiB 3 seconds ago    2 seconds ago
linuxuser@linuxuser:~/rkt-example$ sudo rkt --insecure-options=image run hello-0.0.1-linux-amd64.aci
```



```
linuxuser@linuxuser:~/Downloads$ sudo dpkg -i rkt_1.24.0-1_amd64.deb
[sudo] password for linuxuser:
Selecting previously unselected package rkt.
(Reading database ... 238861 files and directories currently installed.)
Preparing to unpack rkt_1.24.0-1_amd64.deb ...
Unpacking rkt (1.24.0-1) ...
Setting up rkt (1.24.0-1) ...
Processing triggers for man-db (2.7.5-1) ...
linuxuser@linuxuser:~/Downloads$
```

01 Install rkt

First, download the relevant distro version from the rkt release page on GitHub (**github.com/coreos/rkt/tree/master/rkt**). We installed the Debian package. In this case, download the DEB file and navigate to the downloaded file directory using the terminal. Then execute:

```
$ sudo dpkg -i rkt_1.24.0-1_amd64.deb
```

The rkt daemon starts automatically. To install rkt from a zipped file, download the tar.gz file, navigate to the directory from the terminal, and then execute:

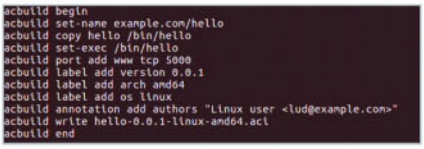
```
$ tar xfv rkt-v1.24.0.tar.gz
```

If you want the rkt source code, download the source code ZIP file from the same release location.

02 Docker architecture

Create your own application in any preferred language. For this example, we will use Go. We create a file called **hello.go**:

```
$ vi hello.go
package main
import (
```



```
acbuild begin
acbuild set-name example.com/hello
acbuild copy hello /bin/hello
acbuild set-exec /bin/hello
acbuild port add www tcp 5000
acbuild label add version 0.0.1
acbuild label add arch amd64
acbuild label add os linux
acbuild annotation add authors "Linux user <lud@example.com>"
acbuild write hello-0.0.1-linux-amd64.aci
acbuild end
```

```
     "log"
)
func main() {
        log.Printf("Hello World")
   }
```

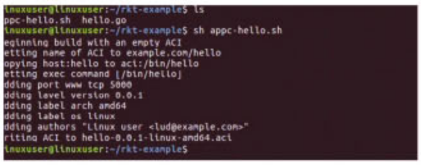
And now we create an image using acbuild in the script file:

```
$ vi appc-hello.sh
acbuild begin
acbuild set-name example.com/hello
acbuild copy hello /bin/hello
acbuild set-exec /bin/hello
acbuild port add www tcp 5000
acbuild label add version 0.0.1
acbuild label add arch amd64
acbuild label add os linux
acbuild annotation add authors
"Linux user <lud@example.com>"
acbuild write hello-0.0.1-linux-
amd64.aci
acbuild end
```

03 Image creation

An image is created when you execute the script you created in the last step.

```
$ ./appc-hello.sh
```



```
inuxuser@linuxuser:~/rkt-example$ ls
ppc-hello.sh  hello.go
inuxuser@linuxuser:~/rkt-example$ sh appc-hello.sh
eginning build with an empty ACI
etting name of ACI to example.com/hello
opying host:hello to aci:/bin/hello
etting exec command [/bin/hello]
dding port www tcp 5000
dding level version 0.0.1
dding label arch amd64
dding label os linux
dding authors "Linux user <lud@example.com>"
riting ACI to hello-0.0.1-linux-amd64.aci
inuxuser@linuxuser:~/rkt-example$
```

Upon executing the script, you will see the logs being generated; they'll look something like this:

```
Beginning build with an empty ACI
Setting name of ACI to example.com/
hello
Copying host:hello to aci:/bin/hello
Setting exec command [/bin/hello]
…
Writing ACI to hello-0.0.1-linux-
amd64.aci
```

Finally, an ACI file is created that is an unsigned, unencrypted appc container image. This can be run with rkt.

04 Run rkt

Check rkt's image list using:

```
$ sudo rkt image list
```

Now launch the container using:

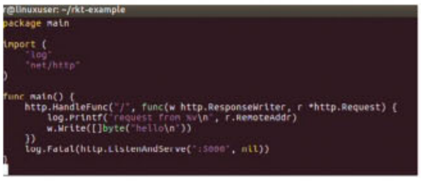
```
$ sudo rkt --insecure-options=image
run hello-0.0.1-linux-amd64.aci
```

By default, rkt expects your images to be signed, so the **--insecure-options** option is required. Now that the app is running, rkt can also run as a daemon. Pass three escape characters (**Ctrl+]**) to stop the container. To clean unused images and containers, execute:

```
$ sudo rkt gc
```

This cleans unused images. For cleaning unused containers, execute:

```
$ sudo rkt image gc
```



```
r@linuxuser: ~/rkt-example
package main

import (
    "log"
    "net/http"
)

func main() {
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        log.Printf("request from %v\n", r.RemoteAddr)
        w.Write([]byte("hello\n"))
    })
    log.Fatal(http.ListenAndServe(":5000", nil))
}
```

05 Test with curl

Create a program that creates a HTTP server listening on port 5000 and prints 'hello' whenever a request is received:

```
$ vi hello.go
package main

import (
    "log"
    "net/http"
)

func main() {
    http.HandleFunc("/", func(w http.
ResponseWriter, r *http.Request) {
        log.Printf("request from %v\n",
r.RemoteAddr)
        w.Write([]byte("hello\n"))
    })
    log.Fatal(http.
ListenAndServe(":5000", nil))
}
```

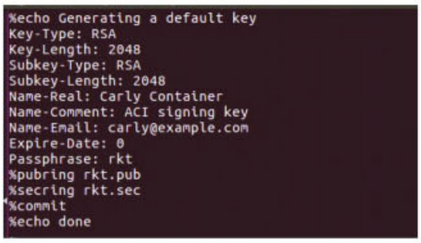
Use **appc-hello.sh** (created in earlier steps) to build it and then launch the container. rkt assigns an IP address to the running container. Find out the IP address using:

```
$ sudo rkt list
```

Then curl the IP on port 5000:

```
$ curl 172.16.28.2:5000
```

This prints 'hello'.



```
%echo Generating a default key
Key-Type: RSA
Key-Length: 2048
Subkey-Type: RSA
Subkey-Length: 2048
Name-Real: Carly Container
Name-Comment: ACI signing key
Name-Email: carly@example.com
Expire-Date: 0
Passphrase: rkt
%pubring rkt.pub
%secring rkt.sec
%commit
%echo done
```

06 Signing images: generate GPG keys

The images that were created in earlier steps were unsigned. If you want to sign the image, use GPG. Create a batch file named gpg-batch:

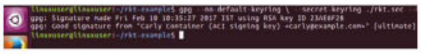
```
%echo Generating a default key
Key-Type: RSA
…
%commit
%echo done
```

Now generate a key using batch mode:

```
$ gpg --batch --gen-key gpg-batch
```

Once the key is created, list the keys and add the relevant key as a trusted key using GPG. Now export the public key:

```
$ gpg --no-default-keyring --armor \
--secret-keyring ./rkt.sec --keyring ./
rkt.pub \
--export lud@example.com > pubkeys.gpg
```



```
linuxuser@linuxuser:~/rkt-example$ gpg --no-default-keyring \ --secret-keyring ./rkt.sec
gpg: Signature made Fri Feb 10 10:35:27 2017 IST using RSA key ID 23AE0F28
gpg: Good signature from "Carly Container (ACI signing key) <carly@example.com>" [ultimate]
linuxuser@linuxuser:~/rkt-example$
```

07 Signing images: signing ACI

Now that you have the public key, sign the ACI file using GPG as follows:

```
$ gpg --no-default-keyring --armor \
--secret-keyring ./rkt.sec --keyring ./
rkt.pub \
--output hello-0.0.1-linux-amd64.aci.
asc \
--detach-sig hello-0.0.1-linux-amd64.aci
```

Verify the signed images:

```
$ gpg --no-default-keyring \
--secret-keyring ./rkt.sec --keyring ./
rkt.pub \
--verify hello-0.0.1-linux-amd64.aci.asc
hello-0.0.1-linux-amd64.aci
```

You will see the signature details on execution.

The files created as part of building and signing the images for our hello.go application are **hello-0.0.1-linux-amd64.aci.asc**, **hello-0.0.1-linux-amd64.aci** and **pubkeys.gpg**.

08 Restrict system resources

Restrict the system resources like CPU and permissible memory that will be used by a container using rkt. Containers inherit resource limits that are configured in the system service unit file. The resource control settings restrict CPU time quota, I/O and memory resource. To restrict CPU time quota, create a unit file:

```
[Service]
ExecStart=/usr/bin/rkt run s-urbaniak.
github.io/images/stress:0.0.1
CPUQuota=20%
```

This ensures that the executed process will not get more than 20 per cent of CPU time on one CPU. To check the usage, use:

```
$ ps -p <PID> -o %cpu%
```

...where <PID> is the process ID for the running container.

09 rkt with systemd

rkt co-operates with init systems like systemd. The latter is used for starting, stopping and checking status for containers. The lifecycle of rkt pods is directly managed by systemd, as it does not interpose a long-running daemon. Create a unit file using rkt:

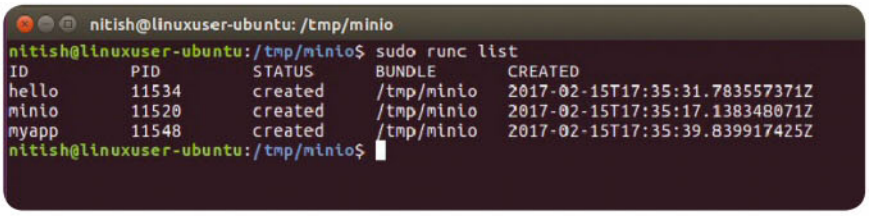
```
$ vi rkt-unit-file
[Unit]
Description=etcd

[Service]
Slice=machine.slice
ExecStart=/usr/bin/rkt run coreos.com/
etcd:v2.2.5
KillMode=mixed
Restart=always
```

This runs an etcd instance under systemd. Manage this using these commands:

```
$ sudo systemctl start etcd.service
$ sudo systemctl stop etcd.service
$ sudo systemctl restart etcd.service
$ sudo systemctl enable etcd.service
$ sudo systemctl disable etcd.service
```

WORK WITH LIGHTWEIGHT STANDALONE CONTAINERS USING **RUNC**

runC is the universal container runtime from the Open Container Initiative



```
nitish@linuxuser-ubuntu: /tmp/minio
nitish@linuxuser-ubuntu:/tmp/minio$ sudo runc list
ID          PID         STATUS      BUNDLE       CREATED
hello       11534       created     /tmp/minio   2017-02-15T17:35:31.783557371Z
minio       11520       created     /tmp/minio   2017-02-15T17:35:17.138348071Z
myapp       11548       created     /tmp/minio   2017-02-15T17:35:39.839917425Z
nitish@linuxuser-ubuntu:/tmp/minio$ 
```

Docker has features that make the sandboxing environment abstract. It abstracts the specifics of the underlying host, without needing to rewrite the application and without excessive performance overheads. These features were integrated into a unified low-level component called runC. Later, Docker Inc made runC as a standalone tool. runC is a lightweight, portable container runtime. It has all the plumbing code that is used by Docker to interact with container-related system features. It is designed for security and scalability and is independent of the rest of the Docker platform. It has native support for security features available in Linux – SELinux, Apparmor, seccomp, control groups, capability drop, pivot_root, uid/gid dropping and more. runC has portable performance profiles, contributed by Google engineers. The Open Container Project, under the auspices of the Linux Foundation, governs it. runC aims to make standard containers available everywhere. Containers start as child processes of runC. They can be embedded into other systems without having a running daemon. runC does not restrict you to a particular workflow or deployment setup: it requires root file system and configurations only. It is based on the battle-tested plumbing used by Docker. In production, you can either use it as a part of a Docker deployment or as a standalone in your custom platform.

01 runC features

runC provides full support for Linux namespaces, including user namespaces. The security features that are available in Linux are also supported by runC. runC provides native support for live migration, Windows 10 containers, and is planning native support for Arm Power, Sparc and hardware features like DODK, sr-iov, tpm and secure enclave. runC is built on libcontainer, which powers Docker Engine installations. Though its configuration standards are formal and governed by the Open Container Project, it does not force you to have a restrictive workflow or deployment setup.

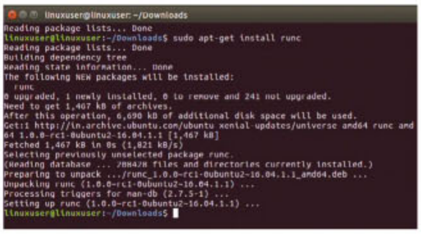
02 Installation

To install runC on your Linux machine, download runc-linux-amd64 from the GitHub releases repository (**github.com/opencontainers/runc**). Install from the package. If you need source code, then download the ZIP file.

```
$ sudo mv runc-linux-amd64 /usr/
bin/runc
$ sudo chmod a+x /usr/bin/runc
```

Or you can directly install from the Ubuntu package list:

```
$ sudo apt-get update
```



```
linuxuser@linuxuser: ~/Downloads
Reading package lists... Done
linuxuser@linuxuser:~/Downloads$ sudo apt-get install runc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  runc
0 upgraded, 1 newly installed, 0 to remove and 241 not upgraded.
Need to get 1,467 kB of archives.
After this operation, 6,690 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 runc amd
64 1.0.0-rc1-0ubuntu2-16.04.1.1 [1,467 kB]
Fetched 1,467 kB in 0s (1,821 kB/s)
Selecting previously unselected package runc.
(Reading database ... 208428 files and directories currently installed.)
Preparing to unpack .../runc_1.0.0-rc1-0ubuntu2-16.04.1.1_amd64.deb ...
Unpacking runc (1.0.0-rc1-0ubuntu2-16.04.1.1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up runc (1.0.0-rc1-0ubuntu2-16.04.1.1) ...
linuxuser@linuxuser:~/Downloads$ 
```

Then execute:

```
$ sudo apt-get install runc
```

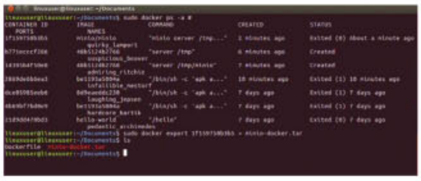
You need superuser access during installation so that you can use the package post-installation.

03 Root file system

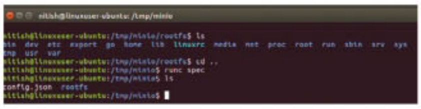
Only the root file system and a configuration are required to start runC containers. runC cannot create or extract the root file system on its own. Use Docker to create it:

```
$ mkdir /tmp/myapp/rootfs
$ docker export myapp | tar xvfC -
/tmp/myapp/rootfs
```

You may use other tools that use OCI images to create the root file system, or use a directory that already has a root file system in it. Check the directory:



```
linuxuser@linuxuser: ~/Documents
linuxuser@linuxuser:~/Documents$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
    PORTS               NAMES
[illegible]         minio/minio         "minio server /tmp..."   2 minutes ago       Exited (0) About a minute ago
                        quirky_lamport
[illegible]         4bb5124b2766        "server /tmp"            6 minutes ago       Created
                        suspicious_beaver
[illegible]         4bb5124b2766        "server /tmp/minio"      7 minutes ago       Created
                        admiring_ritchie
[illegible]         [illegible]         "/bin/sh -c 'apk a..."   10 minutes ago      Exited (1) 10 minutes ago
                        infallible_nestorf
[illegible]         [illegible]         "/bin/sh -c 'apk a..."   7 days ago          Exited (1) 7 days ago
                        laughing_jepsen
[illegible]         [illegible]         "/bin/sh -c 'apk a..."   7 days ago          Exited (1) 7 days ago
                        hardcore_kartik
[illegible]         hello-world         "/hello"                 7 days ago          Exited (0) 7 days ago
                        pedantic_archimedes
linuxuser@linuxuser:~/Documents$ sudo docker export [illegible] > minio-docker.tar
linuxuser@linuxuser:~/Documents$ ls
Dockerfile  minio-docker.tar
linuxuser@linuxuser:~/Documents$ 
```



```
nitish@linuxuser-ubuntu: /tmp/minio
nitish@linuxuser-ubuntu:/tmp/minio/rootfs$ ls
bin  dev  etc  export  go  home  lib  linuxrc  media  mnt  proc  root  run  sbin  srv  sys
tmp  usr  var
nitish@linuxuser-ubuntu:/tmp/minio/rootfs$ cd ..
nitish@linuxuser-ubuntu:/tmp/minio$ runc spec
nitish@linuxuser-ubuntu:/tmp/minio$ ls
config.json  rootfs
nitish@linuxuser-ubuntu:/tmp/minio$ 
```

```
$ ls /tmp/myapp/rootfs
```

Then, in the parent directory, create an OCI runtime configuration file using runC:

```
$ cd /tmp/myapp
$ runc spec
$ ls
```

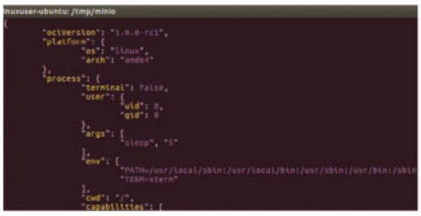
Check that **config.json** is created.

04 Configuration file

Edit the **config.json** file generated in the last step:

```
$ sudo vi config.json
```

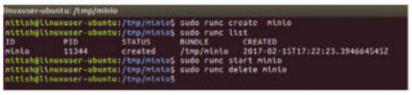
Edit the configurations defined in it or add a



```
nuxuser-ubuntu: /tmp/minio
{
        "ociVersion": "1.0.0-rc1",
        "platform": {
                "os": "linux",
                "arch": "amd64"
        },
        "process": {
                "terminal": false,
                "user": {
                        "uid": 0,
                        "gid": 0
                },
                "args": [
                        "sleep", "5"
                ],
                "env": [
                        "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin
                        "TERM=xterm"
                ],
                "cwd": "/",
                "capabilities": [
```

new configuration as required. By default, an SSH session is created inside the container using **config.json**, which is created by the runC **spec** command. If you want to launch the container in the background, remove the terminal settings from it. Update the process field, for example:

```
"process": {
  "terminal": false,
  "user": {
    "uid": 0,
    "gid": 0
  },
  "args": [
    "sleep", "5"
  ],
  "env": [
    "PATH=/usr/local/sbin:/usr/local/
bin:/usr/sbin:/usr/bin:/sbin:/bin",
  "TERM=xterm"
  ],
…
```



```
linuxuser-ubuntu: /tmp/minio
nitish@linuxuser-ubuntu:/tmp/minio$ sudo runc create  minio
nitish@linuxuser-ubuntu:/tmp/minio$ sudo runc list
ID          PID         STATUS      BUNDLE       CREATED
minio       11344       created     /tmp/minio   2017-02-15T17:22:23.394664545Z
nitish@linuxuser-ubuntu:/tmp/minio$ sudo runc start minio
nitish@linuxuser-ubuntu:/tmp/minio$ sudo runc delete minio
nitish@linuxuser-ubuntu:/tmp/minio$
```

05 Container lifecycles

The **run** command is used to create a runC container. It handles the creation, start and deletion of the container. First, create a container ID:

```
$ cd /tmp/myapp
$ runc create containerId
```

Now create the container:

```
$ runc run containerid
```

View the list of created containers:

```
$ runc list
```

To start the process inside the container:

```
$ runc start containerid
```

To delete the container on exit, execute:

```
$ runc delete mycontainerid
```

This allows higher-level systems to manage runC. This is used to set up the network stack of the container between create and start.

06 Systemd integration

runC does not daemonise to manage containers. It launches directly from the command-line interface, so it integrates with init systems like systemd. For example, the systemd unit file could be something like:

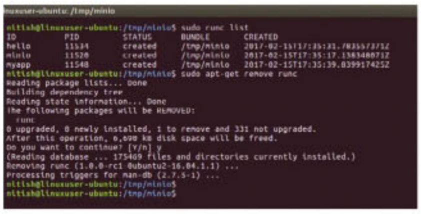
```
$ vi myapp-unit-file
[Unit]
Description=Start My Container

[Service]
Type=forking
ExecStart=/usr/local/sbin/runc run
-d --pid-file /run/containerid.pid
containerid
ExecStopPost=/usr/local/sbin/runc
delete containerid
WorkingDirectory=/tmp/myapp
PIDFile=/run/mycontainerid.pid

[Install]
WantedBy=multi-user.target
```

It can also be used with other init systems and process supervisors to ensure that containers are restarted when they exit.

07 Uninstalling runC

If you want to uninstall runC from your system, execute:

> “runC is designed for security and scalability and is independent of the rest of the Docker platform”



```
nuxuser-ubuntu: /tmp/minio
nitish@linuxuser-ubuntu:/tmp/minio$ sudo runc list
ID          PID         STATUS      BUNDLE       CREATED
hello       11534       created     /tmp/minio   2017-02-15T17:35:31.783557371Z
minio       11520       created     /tmp/minio   2017-02-15T17:35:17.138348071Z
myapp       11548       created     /tmp/minio   2017-02-15T17:35:39.839917425Z
nitish@linuxuser-ubuntu:/tmp/minio$ sudo apt-get remove runc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be REMOVED:
  runc
0 upgraded, 0 newly installed, 1 to remove and 331 not upgraded.
After this operation, 6,690 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 175469 files and directories currently installed.)
Removing runc (1.0.0-rc1 0ubuntu2-16.04.1.1) ...
Processing triggers for man-db (2.7.5-1) ...
nitish@linuxuser-ubuntu:/tmp/minio$
nitish@linuxuser-ubuntu:/tmp/minio$
```

```
$ sudo apt-get remove  runc
```

If you want to remove runC packages along with their dependencies, execute:

```
$ sudo apt-get remove --auto-remove
runc
```

To completely remove runC with all of its configuration files and data, execute:

```
$ sudo apt-get purge runc
```

Or use:

```
$ sudo apt-get purge --auto-remove runc
```

These commands remove all configuration files and data associated with the runC package. They cannot be recovered, so only use them if you are very sure!

08 runC versus Docker

Docker uses technologies like Linux, Go, lxc, aufs, lvm, iptables, VirtualBox, vxlan, mesos, etcd, consul and systemd. Although these components are reusable, they make Docker complex. runC is a lightweight, portable container runtime. It does not depend on the rest of the Docker platform. Although runC takes help from Docker to create a root file system, Docker and runC both support portability, ubiquity and scalability. runC can be used as part of a Docker deployment or standalone distribution.

09 runC vs rkt

runC and rkt are both implementations of the Open Container Initiative specification. Low-level details of the host operating system and configuration are required in runC, whereas rkt does not expect the user to understand the low-level details of the host operating system. Neither have centralised daemons, so they both integrate with init systems like systemd. runC requires container images to be downloaded or cryptographically verified separately. rkt can download Docker images as well as App Container images.

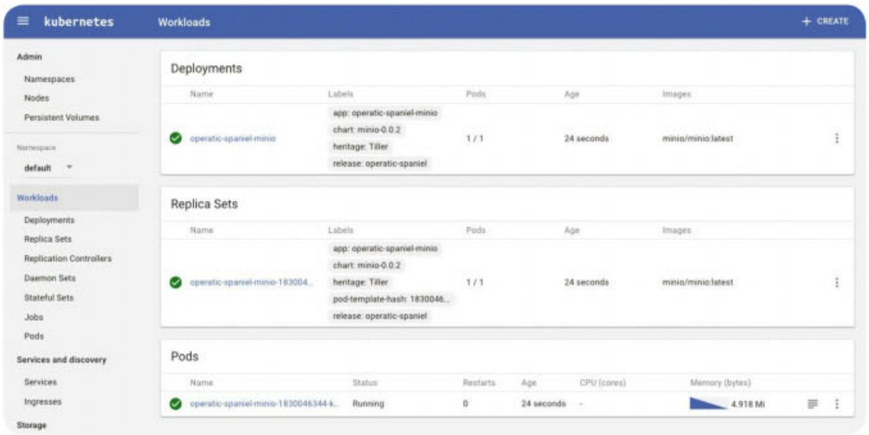
MANAGE AND ORCHESTRATE CONTAINERS WITH **KUBERNETES**

With Kubernetes you can move containers around if there's a hardware problem

Microservice architecture combined with containerised applications offers a great application deployment solution, capable of handling large amounts of traffic. But this solution is still based on an abstraction: the containers are still running on a machine with finite resources. What if one of those machines goes down and you need to move containers?

Container orchestration platforms manage all of the bare-metal or virtual machines that you need to run your containers on. These platforms also manage your containers, launching them on the underlying machines, making sure they are distributed, and keeping them healthy. Kubernetes is one such container orchestration system.

Kubernetes runs all of your application containers using a desired state philosophy. This means that you can say how many instances of an app you need and the system will then ensure that you always have that many replicas running. If one goes down, it restarts another one. If too many are running, it kills the extras. Kubernetes was created by Google, based on its experience of building its own container orchestration system, Borg, over the past ten years. There are a lot of components that make Kubernetes work; let's take a look at them.



Deployments

Name	Labels	Pods	Age	Images
operatic-spaniel-minio	app: operatic-spaniel-minio chart: minio-0.0.2 heritage: Tiller release: operatic-spaniel	1 / 1	24 seconds	minio/minio:latest

Replica Sets

Name	Labels	Pods	Age	Images
operatic-spaniel-minio-183004...	app: operatic-spaniel-minio chart: minio-0.0.2 heritage: Tiller pod-template-hash: 1830046... release: operatic-spaniel	1 / 1	24 seconds	minio/minio:latest

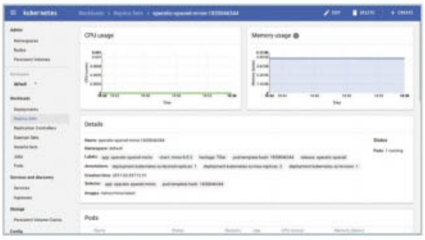
Pods

Name	Status	Restarts	Age	CPU (cores)	Memory (bytes)
operatic-spaniel-minio-1830046344-k...	Running	0	24 seconds	-	4.918 Mi

01 Containers and pods

Pods are the smallest unit of execution in Kubernetes, while containers are subatomic components. That means you never just run a container: it always runs inside a control structure known as a pod. But this doesn't mean you can't run Docker images based on Dockerfiles exactly the same way you do in Docker. In this case, Kubernetes will spin up a single container pod.

The other Kubernetes components spin up one or more pods, or connect one or more pods to the network. Pods are composed of one or more containers. Containers running in the same pod share disk, localhost, security context and other properties.

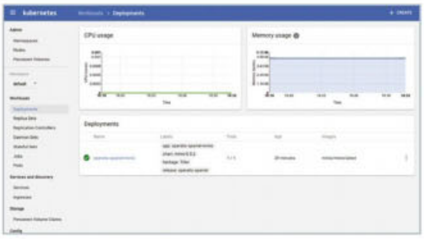


02 ReplicaSet

A ReplicaSet is one of the core parts of the Kubernetes system. It ensures that a specified number of pod replicas are running at any one time. If there are too many pods, ReplicaSet will kill some. If there are too few, it will start more. Unlike manually created pods, the pods maintained by a ReplicaSet are automatically replaced if they fail, are deleted or are terminated.

03 Deployment

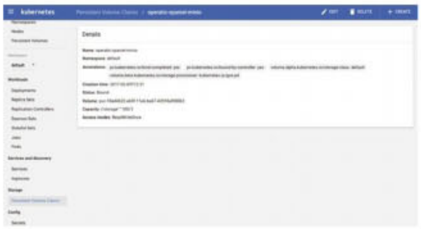
A deployment can be thought of as an abstraction containing pods and a ReplicaSet. When you create a deployment, a ReplicaSet and pods are brought up. You only need to describe the desired state of a deployment object, and the deployment controller will change the actual state to the desired state at a controlled rate for you. You can define deployments by creating new resources, or replace existing ones with new ones. The recommended way to deploy an application on Kubernetes is by creating a deployment.



04 StatefulSet

Many distributed and stateful applications need to have unique, pre-known network identifiers before they are deployed. Stateful applications also need stable, persistent storage that is available to be bound to the appropriate pod even after the previous pod has gone down. StatefulSets provide all this to an

application. For example, in a StatefulSet with n replicas, each pod in the StatefulSet will be assigned an integer ordinal in the range [0-n], which is unique over the Set.

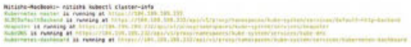


05 PersistentVolumeClaim

A stateful application can request storage using PersistentVolumeClaims (PVCs). PVCs are similar to pods. Just like pods consume node resources and can request specific levels of resources (CPU and memory), PVCs consume persistent volume resources and can request specific size and access modes (eg they can be mounted once to read/write or many times in a read-only state). PVCs allow users to consume abstract storage resources while the responsibility to provide the resources lies with the platform.

06 Services

Services connect ephemeral pods to internal or external processes that need to be long-running, like an API endpoint. Suppose you have three pods running a web server container, and you need a way to route requests from the public internet to your containers. You can do this by setting up a service that uses a load balancer to route requests from a public IP address to one of the containers. Give the pods a label, say ‘web_server’, and then in the service definition say: ‘Serve port 80 using any container labelled web_server.’



```
Nitishs-MacBook:~ nitish$ kubectl cluster-info
Kubernetes master is running at https://104.198.185.232
GLBCDefaultBackend is running at https://104.198.185.232/api/v1/proxy/namespaces/kube-system/services/default-http-backend
Heapster is running at https://104.198.185.232/api/v1/proxy/namespaces/kube-system/services/heapster
KubeDNS is running at https://104.198.185.232/api/v1/proxy/namespaces/kube-system/services/kube-dns
kubernetes-dashboard is running at https://104.198.185.232/api/v1/proxy/namespaces/kube-system/services/kubernetes-dashboard
```

07 The kubectl tool

To deploy and manage applications on Kubernetes, you can use the Kubernetes command-line tool, kubectl. It lets you inspect your cluster resources, create, delete and update components, and much more. Install kubectl using:

```
$ curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
```

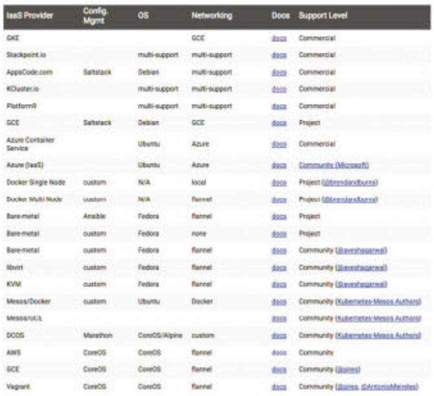
You should use the following syntax to run kubectl commands from your terminal window:

```
$ kubectl [command] [TYPE] [NAME] [flags]
```

…where **command** specifies the operation you want to perform, **TYPE** specifies the resource type, **NAME** specifies the resource name, and **flags** are additional inputs to the command.

08 Pick the right platform

Kubernetes doesn’t have any prerequisites for installation; you can install it on a range of platforms, from a laptop to virtual machines, to cloud providers to bare-metal servers. The only thing to be aware of here is that the capacity of your cluster is based on the underlying platform. For users looking to get acquainted with Kubernetes, the best way to get started is MiniKube, a local Docker-based solution. This creates a single-node Kubernetes setup on your laptop. For cloud providers like GCE and AWS, setting up Kubernetes is easy. While GCE provides Kubernetes support out of the box, you can use the kops tool for AWS.



IaaS Provider	Config. Mgmt	OS	Networking	Docs	Support Level
GKE			GCE	docs	Commercial
Stackpoint.io		multi-support	multi-support	docs	Commercial
AppsCode.com	Saltstack	Debian	multi-support	docs	Commercial
KCluster.io		multi-support	multi-support	docs	Commercial
Platform9		multi-support	multi-support	docs	Commercial
GCE	Saltstack	Debian	GCE	docs	Project
Azure Container Service		Ubuntu	Azure	docs	Commercial
Azure (IaaS)		Ubuntu	Azure	docs	Community (Microsoft)
Docker Single Node	custom	N/A	local	docs	Project (@brendandburns)
Docker Multi Node	custom	N/A	flannel	docs	Project (@brendandburns)
Bare-metal	Ansible	Fedora	flannel	docs	Project
Bare-metal	custom	Fedora	none	docs	Project
Bare-metal	custom	Fedora	flannel	docs	Community (@aveshagarwal)
libvirt	custom	Fedora	flannel	docs	Community (@aveshagarwal)
KVM	custom	Fedora	flannel	docs	Community (@aveshagarwal)
Mesos/Docker	custom	Ubuntu	Docker	docs	Community (Kubernetes-Mesos Authors)
Mesos/GCE				docs	Community (Kubernetes-Mesos Authors)
DCOS	Marathon	CoreOS/Alpine	custom	docs	Community (Kubernetes-Mesos Authors)
AWS	CoreOS	CoreOS	flannel	docs	Community
GCE	CoreOS	CoreOS	flannel	docs	Community (@pires)
Vagrant	CoreOS	CoreOS	flannel	docs	Community (@pires, @AntonioMeireles)

09 Deploy applications on Kubernetes

Now that you have an idea of various Kubernetes constructs and how they work, we’ll see how to actually deploy an application. To do so, you need to create a configuration file in a format defined by Kubernetes. Then, you can use the **kubectl create** command and pass the configuration file to Kubernetes. The command takes this format:

```
$ kubectl create -f config.yaml
```

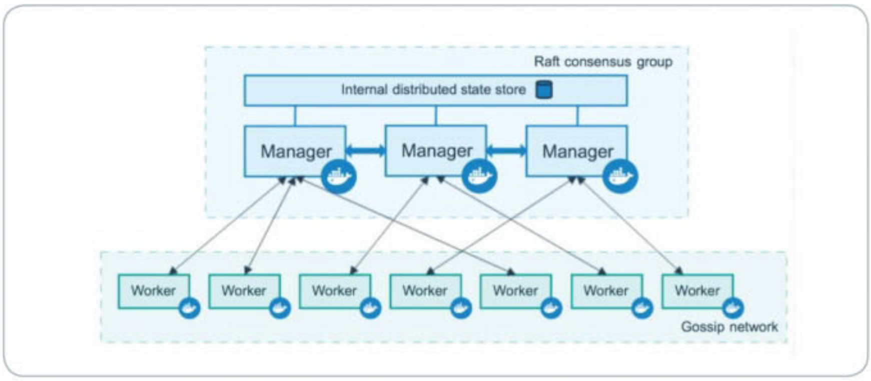
Another option is to use the Kubernetes package manager, Helm. The package manager maintains a public repository of Kubernetes configurations called Helm charts. To deploy an application via Helm, you can use the **helm install** command with the chart name. For example, to install the Minio chart, you can use:

```
$ helm install stable/minio
```

> “Container orchestration platforms like Kubernetes manage all of the bare-metal or virtual machines that you need to run your containers on”

NATIVELY MANAGE A CLUSTER WITH **DOCKER SWARM**

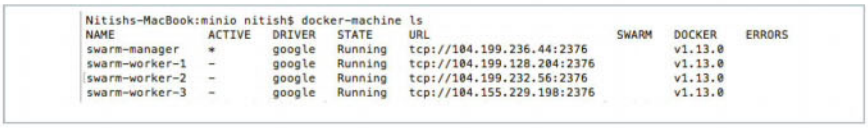
Why use one container when you can run a whole bunch of them concurrently?



While projects like Kubernetes and DC/OS (dcos.io) are building container orchestration platforms that can work with different container formats, Docker Inc integrates container orchestration with its core offering, the Docker Engine. As of Docker Engine v1.12.0, Swarm is built-in and can be used without the need to install any extra software. You will need networked host machines to run the container cluster, though.

This integration of Swarm with Docker Engine means you can natively manage a cluster of Docker Engines. This means you can use the Docker CLI to create a swarm, deploy application services to a swarm and manage swarm behaviour. With the latest Docker Engine v1.13.0, you can even use Docker Compose files to manage your Docker swarm. For the uninitiated, Docker Compose is a tool for defining and running multi-container Docker applications on a single node. Compose works on files called **docker-compose.yml**. As of Docker Engine v1.13.0, these Compose files can be used to configure a multi-node swarm as well.

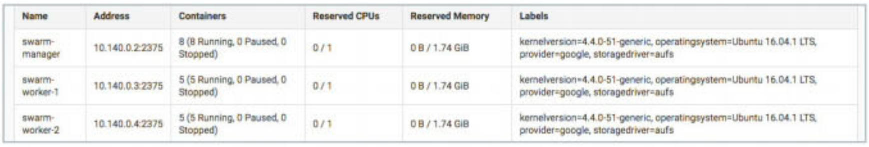
Swarm provides all the production-ready features you need, including load balancing, service discovery, rolling updates, scaling and more. We'll learn about all these features in a little while. For now, just keep in mind that when you run Docker without using swarm mode, you execute container commands. When you run Docker in swarm mode, you orchestrate services.



```
Nitishs-MacBook:minio nitish$ docker-machine ls
NAME             ACTIVE   DRIVER   STATE     URL                            SWARM   DOCKER    ERRORS
swarm-manager    *        google   Running   tcp://104.199.236.44:2376              v1.13.0
swarm-worker-1   -        google   Running   tcp://104.199.128.204:2376             v1.13.0
swarm-worker-2   -        google   Running   tcp://104.199.232.56:2376              v1.13.0
swarm-worker-3   -        google   Running   tcp://104.155.229.198:2376             v1.13.0
```

01 Swarm manager node

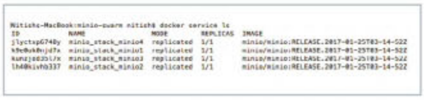
Instances of Docker Engine participating in a swarm are called nodes. A swarm can be of two types of nodes, manager and worker. Manager nodes are the controlling authorities in a swarm and handle all the administrative tasks, like dispatching tasks to a worker and performing the orchestration and cluster management functions required to maintain the desired state of the swarm. When there is more than one manager node in a swarm, they elect a single leader to conduct orchestration tasks. Even when you want to deploy an application to a swarm, you submit a service definition to a manager node.



Name	Address	Containers	Reserved CPUs	Reserved Memory	Labels
swarm-manager	10.140.0.2:2375	8 (8 Running, 0 Paused, 0 Stopped)	0 / 1	0 B / 1.74 GiB	kernelversion=4.4.0-51-generic, operatingsystem=Ubuntu 16.04.1 LTS, provider=google, storagedriver=aufs
swarm-worker-1	10.140.0.3:2375	5 (5 Running, 0 Paused, 0 Stopped)	0 / 1	0 B / 1.74 GiB	kernelversion=4.4.0-51-generic, operatingsystem=Ubuntu 16.04.1 LTS, provider=google, storagedriver=aufs
swarm-worker-2	10.140.0.4:2375	5 (5 Running, 0 Paused, 0 Stopped)	0 / 1	0 B / 1.74 GiB	kernelversion=4.4.0-51-generic, operatingsystem=Ubuntu 16.04.1 LTS, provider=google, storagedriver=aufs

02 Swarm worker nodes

Worker nodes are also instances of Docker Engine whose sole purpose is to execute containers. Worker nodes don't participate in making scheduling decisions, or serve the swarm mode HTTP API. You can create a swarm of one manager node, but you cannot have a worker node without at least one manager node. By default, all managers are also workers. In a single manager node cluster, you can run commands like **docker service create** and the scheduler will place all tasks on the local Engine.



```
Nitishs-MacBook:minio-swarm nitish$ docker service ls
ID            NAME                MODE        REPLICAS  IMAGE
jlyctxpG748y  minio_stack_minio4  replicated  1/1       minio/minio:RELEASE.2017-01-25T03-14-52Z
k9e0uk0njd7x  minio_stack_minio1  replicated  1/1       minio/minio:RELEASE.2017-01-25T03-14-52Z
kunzjsd35l7x  minio_stack_minio3  replicated  1/1       minio/minio:RELEASE.2017-01-25T03-14-52Z
lh40kivhb337  minio_stack_minio2  replicated  1/1       minio/minio:RELEASE.2017-01-25T03-14-52Z
```

03 Swarm services

To deploy an application image in Docker swarm mode, you need to create a service. A service generally includes an image for a microservice within the context of some larger application. Examples of services include an HTTP server, a database, or any other type of executable program that you wish to run in a distributed environment.

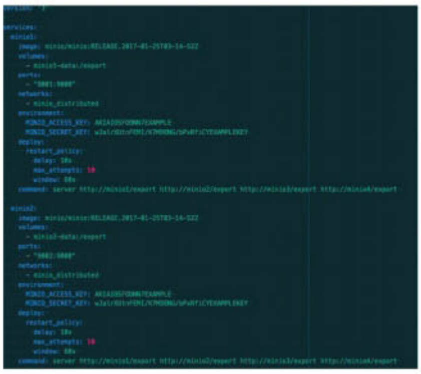
When you deploy a service to a swarm, the swarm manager accepts your service definition as the desired state for the service. Then it schedules the service on nodes in the swarm as one or more replica tasks. The tasks run independently of each other on nodes in the swarm.

04 Deploy services via Compose file

As of the latest Docker engine (v1.13.0) and Docker Compose v3.0, there is cross-compatibility between Compose and the Docker Engine's swarm mode.

> “When you run Docker without using swarm mode, you execute container commands. When you run it in swarm mode, you orchestrate services”

This means you can create a file using the Docker Compose format and use it to deploy services in swarm mode. Not only does this ease the deployment process, it also makes it possible to easily pass along swarm service definitions.



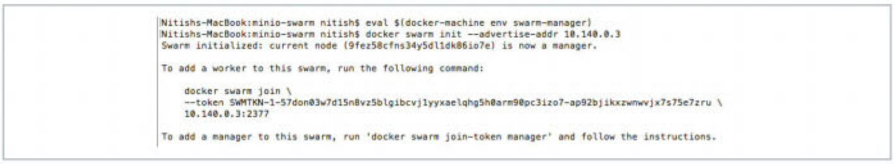
```
version: "3"

services:
  minio1:
    image: minio/minio:RELEASE.2017-01-25T03-14-52Z
    volumes:
      - minio1-data:/export
    ports:
      - "9001:9000"
    networks:
      - minio_distributed
    environment:
      MINIO_ACCESS_KEY: AKIAIOSFODNN7EXAMPLE
      MINIO_SECRET_KEY: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
    deploy:
      restart_policy:
        delay: 10s
        max_attempts: 10
        window: 60s
    command: server http://minio1/export http://minio2/export http://minio3/export http://minio4/export

  minio2:
    image: minio/minio:RELEASE.2017-01-25T03-14-52Z
    volumes:
      - minio2-data:/export
    ports:
      - "9002:9000"
    networks:
      - minio_distributed
    environment:
      MINIO_ACCESS_KEY: AKIAIOSFODNN7EXAMPLE
      MINIO_SECRET_KEY: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
    deploy:
      restart_policy:
        delay: 10s
        max_attempts: 10
        window: 60s
    command: server http://minio1/export http://minio2/export http://minio3/export http://minio4/export
```

05 Compose file format

A Compose file starts with the version section. For a swarm-compatible Compose file, the version should be 3. Next, you need to define the services to be created in the swarm under the services section. Each service starts with the service name followed by the image section. Here you need to specify the image to be used for the service. Next you need to add volumes, ports, networks, environment and deploy sections. Finally, the command section indicates the commands to be executed to launch the service.

06 Load balancing

The swarm manager uses ingress load balancing to expose the services you want to make available externally to the swarm. The swarm manager can automatically assign the service a PublishedPort or you can configure a PublishedPort for the service. You can specify any unused port. In Compose files, the ports section specifies port mapping. If you do not specify a port, the swarm manager assigns the service a port in the 30000-32767 range.



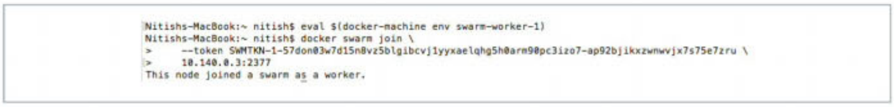
```
Nitishs-MacBook:minio-swarm nitish$ eval $(docker-machine env swarm-manager)
Nitishs-MacBook:minio-swarm nitish$ docker swarm init --advertise-addr 10.140.0.3
Swarm initialized: current node (9fez58cfns34y5dl1dk86io7e) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join \
    --token SWMTKN-1-57don03w7d15n8vz5blgibcvj1yyxaelqhg5h0arm90pc3izo7-ap92bjikxzwnwvjx7s75e7zru \
    10.140.0.3:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
```

07 Create a swarm

To create a swarm, you need Docker Engine v1.12 or higher installed on all the host machines, with the Docker Engine daemon running on all the machines. Also, if a network connects these host machines, the ports 2377 (for cluster management), 7946 (for inter-node communication) and 4789 (for overlay network creation) should be available. Once you have the hosts ready, SSH into the machine you'd like to use as the manager, and initialise the swarm with

```
$ docker swarm init --advertise-addr <MANAGER-IP>
```

Here, you need to set the IP address of the manager machine.

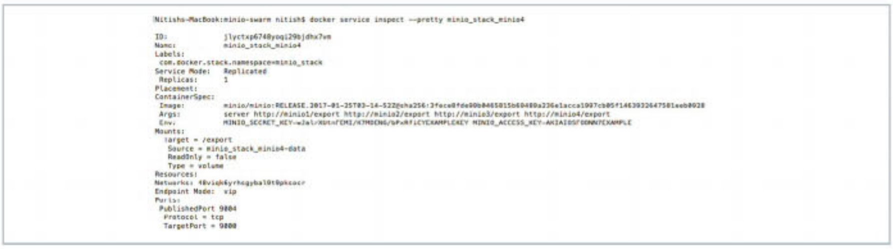


```
Nitishs-MacBook:~ nitish$ eval $(docker-machine env swarm-worker-1)
Nitishs-MacBook:~ nitish$ docker swarm join \
>     --token SWMTKN-1-57don03w7d15n8vz5blgibcvj1yyxaelqhg5h0arm90pc3izo7-ap92bjikxzwnwvjx7s75e7zru \
>     10.140.0.3:2377
This node joined a swarm as a worker.
```

08 Add nodes to a swarm

Once you've created a swarm with a manager node, you're ready to add worker nodes. Open a terminal and SSH into the machine where you want to run a worker node. After logging in to the machine, run the command produced by the **docker swarm init** output from the previous step to create a worker node joined to the existing swarm. Alternatively, you can use the command:

```
$ docker swarm join-token worker
```



```
Nitishs-MacBook:minio-swarm nitish$ docker service inspect --pretty minio_stack_minio4

ID:             jlyctxp6740yoqi29bjdhx7vm
Name:           minio_stack_minio4
Labels:
 com.docker.stack.namespace=minio_stack
Service Mode:   Replicated
 Replicas:      1
Placement:
ContainerSpec:
 Image:         minio/minio:RELEASE.2017-01-25T03-14-52Z@sha256:3fece0fde00b0465015b60480a236e1acca1997cb05f146393264758leeb0028
 Args:          server http://minio1/export http://minio2/export http://minio3/export http://minio4/export
 Env.           MINIO_SECRET_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY MINIO_ACCESS_KEY=AKIAIOSFODNN7EXAMPLE
Mounts:
  Target = /export
   Source = minio_stack_minio4-data
   ReadOnly = false
   Type = volume
Resources:
Networks: 48viqh6yrhogybal0t0pkcocr
Endpoint Mode:  vip
Ports:
 PublishedPort 9004
  Protocol = tcp
  TargetPort = 9000
```

09 Inspect a service

You can use the Docker CLI to see all the details about services running in the swarm. If you haven't already, open a terminal and SSH into the machine where you run your manager node. Then, run the command:

```
$ docker service inspect --pretty <SERVICE-ID>
```

...to display the details about a service in an easily readable format. You can use other **docker service** commands to explore the service details. For example:

```
$ docker service ls
```

...shows all the running services.

```
$ docker service ps <SERVICE-ID>
```

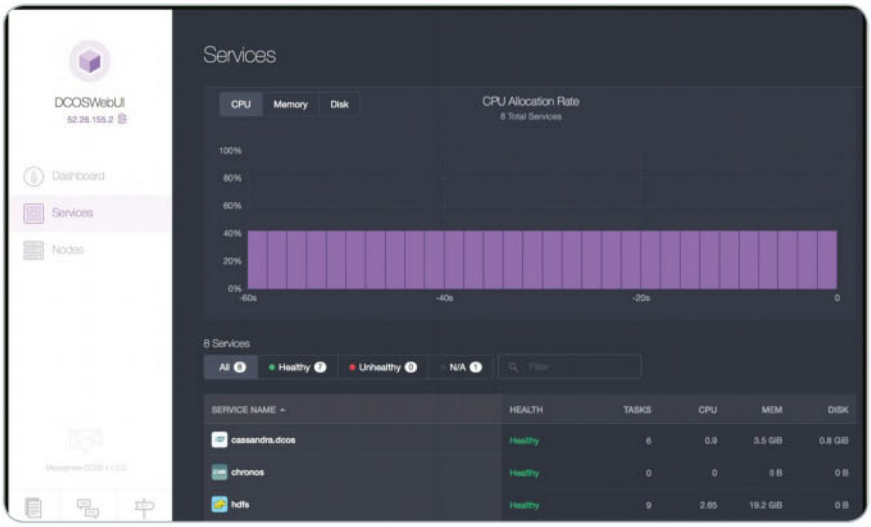
...shows all the tasks running for a service.

RUN ENTERPRISE-LEVEL CONTAINERS WITH MESOSPHERE

This distributed system simplifies building and running applications over large architectures

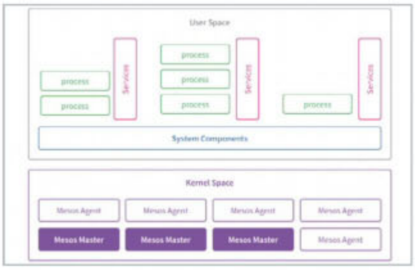
DC/OS is a distributed operating system based on the Apache Mesos distributed systems kernel. It enables the management of multiple machines as if they were a single computer. It automates resource management, schedules process placement, facilitates inter-process communication, and simplifies the installation and management of distributed services. Its web interface and command-line interface facilitate remote management and monitoring of the cluster and its services.

Apache Mesos is the open source distributed systems kernel at the heart of the Mesosphere DC/OS. Mesos began as a research project at UC Berkeley, the birthplace of BSD UNIX. Inspired by Google's proprietary Borg system, the project's goal was to create an open source kernel that simplifies building and running distributed applications at a very large scale and treats the entire data centre as a single giant supercomputer, while also maintaining an extensible architecture.



SERVICE NAME ▲	HEALTH	TASKS	CPU	MEM	DISK
cassandra.dcos	Healthy	6	0.9	3.5 GiB	0.8 GiB
chronos	Healthy	0	0	0 B	0 B
hdfs	Healthy	9	2.65	19.2 GiB	0 B

Traditional monolithic schedulers maintain the complete state of the application and infrastructure underneath, while also performing workload placement logic. This architecture makes it very challenging to scale and even harder to introduce new features and capabilities. With a dual-level architecture, Mesos handles low-level infrastructure scheduling operations, while another layer on top (the framework) handles all the application-specific operations and logic.

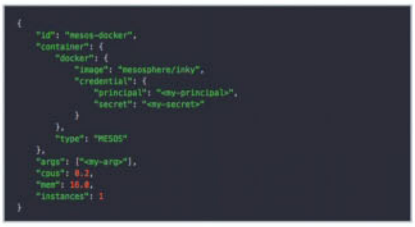


01 Architecture

The DC/OS kernel space is comprised of Mesos masters and Mesos agents. The user space includes system components such as Mesos-DNS, Distributed DNS Proxy, and services such as Marathon and Spark. The user space also includes processes that are managed by the services, for example a Marathon application. The Mesos masters process orchestrates tasks that are run on Mesos agents. It receives resource reports from Mesos agents and distributes those resources to registered DC/OS services, such as Marathon or Spark. Mesos agent nodes run discrete Mesos tasks on behalf of a framework. Private agent nodes run the deployed apps and services through a non-routable network. Public agent nodes run DC/OS apps and services in a publicly accessible network.

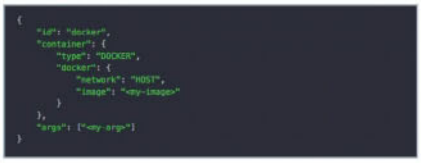
02 Mesos containerisers

Containerisers allow you to run tasks in containers. DC/OS supports two containeriser types – the DC/OS Universal container runtime and Docker containeriser. The Universal container runtime extends the Mesos containeriser to support provisioning Docker container images (AppC is coming soon, too).



```
{
    "id": "mesos-docker",
    "container": {
        "docker": {
            "image": "mesosphere/inky",
            "credential": {
                "principal": "<my-principal>",
                "secret": "<my-secret>"
            }
        },
        "type": "MESOS"
    },
    "args": ["<my-arg>"],
    "cpus": 0.2,
    "mem": 16.0,
    "instances": 1
}
```

This means that you can use both the Mesos containeriser and other container image types in DC/OS. You can also use the Docker container runtime directly with DC/OS, but the Universal container runtime supports running Docker images without depending on the Docker Engine, which allows for better integration with Mesos.



```
{
    "id": "docker",
    "container": {
        "type": "DOCKER",
        "docker": {
            "network": "HOST",
            "image": "<my-image>"
        }
    },
    "args": ["<my-arg>"]
}
```

03 DCOS installation

DC/OS can be installed in the environment of your choice by using a customised setup file or cloud templates. There are detailed instruction steps available for all the major cloud infrastructure providers. You can also use Vagrant to create a cluster of virtual machines on your local machine that can be used for demos, development and testing with DC/OS.

04 Mesos executor and scheduler

Mesos schedulers define new Mesos tasks and assign resources to them (placing them on specific nodes). A scheduler receives resource offers describing CPU, RAM, etc, and allocates them for discrete tasks that can be launched by Mesos agents via executors.

Executor processes are launched and managed by Mesos agents on the agent nodes. Mesos tasks are defined by their scheduler to be run by a specific executor (or the default executor). Each executor runs in its own container.

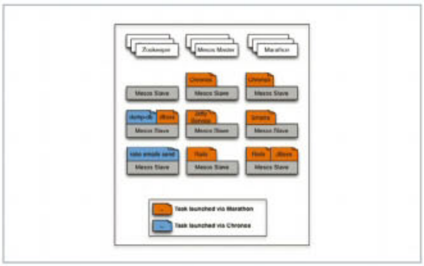
05 Marathon

Marathon is a production-grade container orchestration platform for DC/OS. It is a core component of DC/OS and is the first framework to be bootstrapped once a DC/OS cluster is up. This is because Marathon also runs other Mesos frameworks.

Marathon supports both Mesos containers (using cgroups) and Docker. You can also run Stateful apps by binding persistent storage volumes to your application. This means databases like MySQL, Postgres and object stores like Minio can easily be deployed on DC/OS, and have storage accounted for by the underlying platform.

06 Chronos

Chronos is a Mesos scheduler for running schedule- and dependency-based jobs.



You can think of Chronos as a replacement for cron. It is a distributed and fault-tolerant scheduler that runs on top of Apache Mesos that can be used for job orchestration.

Jobs scheduled with Chronos are configured with ISO8601-based schedules with repeating intervals. Typically, a job is scheduled to run indefinitely, such as once per day or per hour. Dependent jobs may have multiple parents, and will be triggered once all parents have been successfully invoked at least once since the last invocation of the dependent job.

07 Applications

A real-world application like WordPress, Jenkins or any other such applcation can be deployed on DC/OS as a Marathon application. An application is generally a long-running service, with many instances running on multiple hosts. Each application instance is called a task. An application definition describes everything needed to start and maintain the tasks. We'll see how to write an application definition in the following step.

08 Application configuration

Applications can be defined as JSON files in DC/OS. Let us take an example of a trivial application that prints the phrase 'Hello Marathon'. Create a JSON file and add the following details to it:

```
{
    "id": "basic-0",
    "cmd": "while [ true ] ; do echo
'Hello Marathon' ; sleep 5 ; done",
    "cpus": 0.1,
    "mem": 10.0,
    "instances": 1
}
```

The example above is the command that gets executed. Its value is wrapped by the underlying Mesos executor via **/bin/sh -c ${cmd}**. Then you can add the service to DC/OS:

```
$ dcos marathon app add <your-service-
name>.json
```

Once this is done, Marathon hands over execution to Mesos. Mesos creates a sandbox directory for each task. The sandbox directory is a directory on each agent node that acts as an execution environment and contains relevant log files.

09 Using resources in applications

To run non-trivial applications that need resources like files, Marathon has the concept of URIs (uniform resource identifiers). URIs use the Mesos fetcher to do the legwork in terms of downloading (and potentially) extracting resources.

```
{
    "id": "basic-1",
    "cmd": "`chmod u+x cool-script.sh
&& ./cool-script.sh`",
    "cpus": 0.1,
    "mem": 10.0,
    "instances": 1,
    "uris": [
      "https://example.com/app/cool-
script.sh"
    ]
}
```

The example above executes the command, downloads the **cool-script.sh** resource (via Mesos) and makes it available in the service instance's Mesos sandbox. It's straightforward to run applications that use Docker images, so why not put it to the test? For example, this is how to create a simple Docker app in DC/OS:

```
{
  "id": "basic-3",
  "cmd": "python3 -m http.server
8080",
  "cpus": 0.5,
  "mem": 32.0,
  "container": {
    "type": "DOCKER",
    "docker": {
      "image": "python:3",
      "network": "BRIDGE",
      "portMappings": [
        { "containerPort": 8080,
"hostPort": 0 }
      ]
    }
  }
}
```

LinuxUser & Developer | Subscription offer

~~£6.49~~

£4.87 per issue

Subscribe and save 25%*

Every issue, delivered straight to your door

✓

Never miss an issue
13 issues a year, and you'll be sure to get every single one

✓

Delivered to your home
Free delivery of every issue, direct to your doorstep

✓

Get the biggest savings
Get your favourite magazine for less by ordering direct

What our readers are saying about us…

"I've only just found out about this magazine today. It's absolutely brilliant and exactly what I was looking for.
I'm amazed!"
Donald Sleighthholme via Facebook

"@LinuxUserMag just arrived by post. Wow what a fantastic issue! I was just about to start playing with mini-pcs and a soldering iron. TY"
@businessBoris via Twitter

"Thanks for a great magazine. I've been a regular subscriber now for a number of years."
Matt Caswell via email

LinuxUser & Developer | Subscription offer

Pick the subscription that's right for you

MOST FLEXIBLE

Subscribe and save 25%

✔ Automatic renewal – never miss an issue

✔ Pay by Direct Debit

Recurring payment of £29.20 every six issues, saving 25% on the retail price

DIRECT Debit

Instruction to your Bank or Building Society to pay by Direct Debit

Originator's

5 0 1 8 8 4

Name of bank

Address of bank

Account Name

Postcode

Sort Code

Account no

Please pay Imagine Publishing Limited Direct Debits from the account detailed in this instruction subject to the safeguards assured by the Direct Debit guarantee. I understand that this instruction may remain with Imagine Publishing Limited and, if so, details will be passed on electronically to my Bank/Building Society. Banks & Building Societies may not accept Direct Debit instructions for some types of account

Signature

Date

GREAT VALUE

One year subscription

✔ Great offers, available world-wide

✔ One payment, by card or cheque

A simple one-off payment ensures you never miss an issue for one full year. That's 13 issues, direct to your doorstep

□ UK £67.50 (saving 20% on the retail price)

□ Europe £76 □ USA £87 □ Rest of the world £87

Pay by card or cheque

Pay by Credit or Debit card

□ VISA Visa □ MasterCard Mastercard □ AMERICAN EXPRESS Amex

Card number

Expiry date

/

Pay by Cheque

I enclose a cheque for

£

Made payable to Imagine Publishing Ltd

Signature

Date

Your information

Name

Telephone number

Mobile number

Email address

Please tick if you do not wish to receive any promotional material from Imagine Publishing Ltd
□ By post □ By telephone □ By email

Please tick if you do not wish to receive any promotional material from other companies □ By post □ By telephone
□ Please tick if you DO wish to receive such information by email

Address

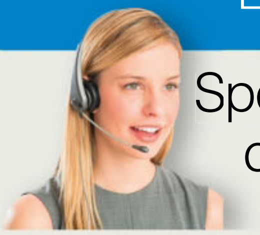
Postcode

Please post this form to

Linux User & Developer Subscriptions, **800 Guillat Avenue, Kent Science Park, Sittingbourne, Kent ME9 8GU**

Order securely online **www.imaginesubs.co.uk/lud**

Enter the promo code ***PS17*** to get these great offers



Speak to one of our friendly customer service team

Call **0844 249 0282**

These offers will expire on
Wednesday 31 May 2017

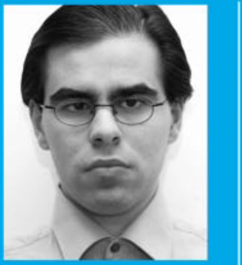
Please quote code PS17

Calls cost 7p per minute plus your telephone company's access charge

*This offer entitles new UK Direct Debit subscribers to pay only £29.20 every 6 issues. Offer code PS17 must be quoted to receive this special subscription price. New subscriptions will start from the next available issue. Details of the Direct Debit guarantee are available on request. Subscribers can cancel this subscription at any time. This offer expires 31st May 2017.

Create a custom image of OpenWRT

Creating a custom build of OpenWRT enables you to take intimate control of the content of your router's firmware



Tam Hanna

Tam's experience in the creation of embedded systems instilled him with a deep interest in build systems. Even though OpenWRT's build system cannot stand up to Yocto's, its simplicity nevertheless makes for some interesting learning!

Resources

OpenWRT
openwrt.org

Tutorial files available:
filesilo.co.uk

The end of the last tutorial on OpenWRT was somewhat discouraging: as the OpenWRT team did not populate many of the packages, our experiments on the Raspberry Pi were severely limited due to the lack of available products.

Fortunately, solving this problem is not particularly difficult. OpenWRT, after all, is but another version of Linux – when this tutorial's author was a young cadet, compiling your own kernel was a badge of honour for every professional user. When this is done, the amount of software available to our OpenWRT build is limited only by our imagination and the memory of the target device.

The following guide will show you how to create your own OpenWRT image for Raspberry Pi process computers. Be aware that this operation is not resource-cheap: the following project was written on an AMD octa-core workstation equipped with SSD flash memory, 16GB of RAM and a cooling system optimised for permanent high performance. Performing the following steps on notebooks – HP's compact sub-notebook families such as the 2540p are common suspects – is likely to lead to thermal issues that could damage your hardware.

As in most other operations, creating a working directory is a great first step. In the case of OpenWRT, be careful: the build system does not allow spaces or weird characters in its path. We used the following working directory:

```
tamhan@TAMHAN14:~/wrtspace$ pwd
/home/tamhan/wrtspace
```

Compiling complex operating systems requires the presence of a variety of tools on the host. The installation instructions recommend Ubuntu users to execute the following commands:

```
sudo apt-get install git-core build-essential
libssl-dev libncurses5-dev unzip gawk zlib1g-dev
sudo apt-get install subversion mercurial
```

In the next step, code must be downloaded from OpenWRT's repositories. This is accomplished by git – due to the limited speed of the service, be prepared to wait a minute or two:

```
tamhan@TAMHAN14:~/wrtspace$ git clone https://
github.com/openwrt/openwrt.git
Cloning into 'openwrt'...
. . .
Receiving objects: 100% (360595/360595), 132.89
MiB | 1.32 MiB/s, done.
. . .
```

Our downloaded code must be expanded by a few external modules. This is best accomplished by changing into the OpenWRT directory, where you must then run two additional commands that download optional packages from various sources:

```
cd openwrt
./scripts/feeds update -a
./scripts/feeds install -a
```

Deploy, non-stop!

Our source blob is not specific to any architecture: it is suitable to compile products for any and every type of device supported by OpenWRT. Our next job involves the creation of a custom configuration for the OpenWRT target device we want to use.

The actual customisation progress can be handled by one of three different configuration utilities. This example will use the menuconfig tool, which old hands of the Linux world should already know from compiling kernels for the desktop. Getting started with it involves entering the following command, which will make OpenWRT start the preparation process:

```
tamhan@TAMHAN14:~/wrtspace/openwrt$ make
menuconfig
```

After about three or four minutes, a window similar to that shown in **Figure 1** pops up.

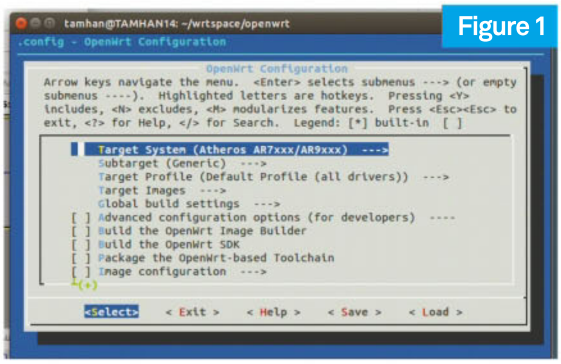


Figure 1

```
tamhan@TAMHAN14: ~/wrtspace/openwrt
.config - OpenWrt Configuration
                        OpenWrt Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
 submenus ----).  Highlighted letters are hotkeys.  Pressing <Y>
 includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to
 exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]

         Target System (Atheros AR7xxx/AR9xxx)  --->
         Subtarget (Generic)  --->
         Target Profile (Default Profile (all drivers))  --->
         Target Images  --->
         Global build settings  --->
     [ ] Advanced configuration options (for developers)  ----
     [ ] Build the OpenWrt Image Builder
     [ ] Build the OpenWrt SDK
     [ ] Package the OpenWrt-based Toolchain
     [ ] Image configuration  --->
     (+)

      <Select>    < Exit >    < Help >    < Save >    < Load >
```

Above Any similarities to SlackWare's installation are purely coincidental

The up and down arrow keys allow you to move the selection; pushing **Enter** allows you activate the option highlighted at the very bottom of the screen. First, select the target system option and set it to BCM27xx.

In the next step, the line below it will allow you to select the exact processor type. The correct value depends on the SoC found in your Raspberry Pi – the table below provides an overview.

Processor type	Board types
BCM2708	Raspberry Pi Model B Raspberry Pi Model B+ Raspberry Pi Compute Module
BCM2709	Raspberry Pi 2 Model B
BCM2710	Raspberry Pi 3 Model B

Fortunately, selecting the wrong value is not particularly difficult – if you pick the wrong type, your target board will not be shown in the target profile selection. When target system, subtarget and target profile have all been set up correctly, use the side keys to enable the Save option and press **Enter**. For the following steps, we assume that the file has been stored as **.config**. The output of the menuconfig tool will prompt you to run Make – but we're not quite there yet:

```
tamhan@TAMHAN14:~/wrtspace/openwrt$ make
menuconfig
```

```
*** End of the configuration.
*** Execute 'make' to start the build or try
'make help'.
```

The OpenWRT team, furthermore, suggests that the defconfig command should be run before invoking the actual compile process:

```
tamhan@TAMHAN14:~/wrtspace/openwrt$ make
defconfig
#
# configuration written to .config
#
```

The defconfig tool uses the information found in the **.config** file, which specifies the target architecture. It is then expanded with a few default parameters, leading to a 'sensible basic configuration'.

In the next step, simply enter **`make`** to start the compile process. Owners of multicore systems would be well advised to pass in the **`-j`** parameter, which informs the Make utility that more than one core should be used.

In that case, however, be aware that the utilisation reached is likely to be relatively low – OpenWRT's compile process is not nearly as parallelised as the one found in systems like Yocto.

With that out of the way, it is time to give your workstation a few minutes' worth of time: our octa-core took about one hour to compile.

When done, the results are presented in the **/bin** folder. Careful observers will note that the path name of the binary folder does not match the one expected – the Raspberry Pi 3 uses a BCM2710 processor.

```
tamhan@TAMHAN14:~/wrtspace/openwrt/bin/
brcm2708$ ls
md5sums                                 packages
openwrt-brcm2708-bcm2710-rpi-3-ext4-sdcard.img
sha256sums
openwrt-brcm2708-bcm2710-rpi-3-ext4-sdcard.img.gz
```

Beware of compile errors

The OpenWRT team explicitly warns developers that multithreaded compiles can run into odd problems. If your compile process fails for inexplicable reasons, invoking **`make`** without **`-j`** can be a nice alternative. Further information on this topic can be found at **bit.ly/2ki27OF**.

> "The amount of software available to our OpenWRT build is limited only by our imagination and the memory of the target device"

An experienced Raspberry Pi jockey should not have much of a problem deploying the image: burn it to an SD card of choice, ram it into your process computer and power up. Actual interaction between your OpenWRT image and the process computer can be accomplished either by SSH or via keyboard. In the next step, try to invoke GCC. Sadly, the process will still fail – the package is not available.

Fight the package wars

OpenWRT is made up of a large variety of packages. By default, the configuration setup in menuconfig compiles the most important packages – it would be stupid to compile everything. Fixing the problem of the missing GCC is not particularly difficult.

Return to the window containing the compile process and make sure that you increase the size of the window before invoking the menuconfig tool – if the window is large enough, all options will be shown on the screen. Changing the screen size of an already-open menuconfig is not necessarily a good idea: we got one or two segmentation faults.

A careful look at the figure below tells us that the various package groups are shown below the image configuration header. Start out by opening the Development menu. It will look a lot like the one shown in **Figure 2**.

A quick glance at the contents of the window reveals groups of brackets next to the packages. Each one of them describes one item, and can take the values <*> and <M>. The difference between these two states is significant: setting something to <*> builds it into the **main.IMG** file, while setting something to <M> creates a package that can be installed by hand.

This difference is not particularly significant in the case of a Raspberry Pi with a 32GB memory card. As mentioned

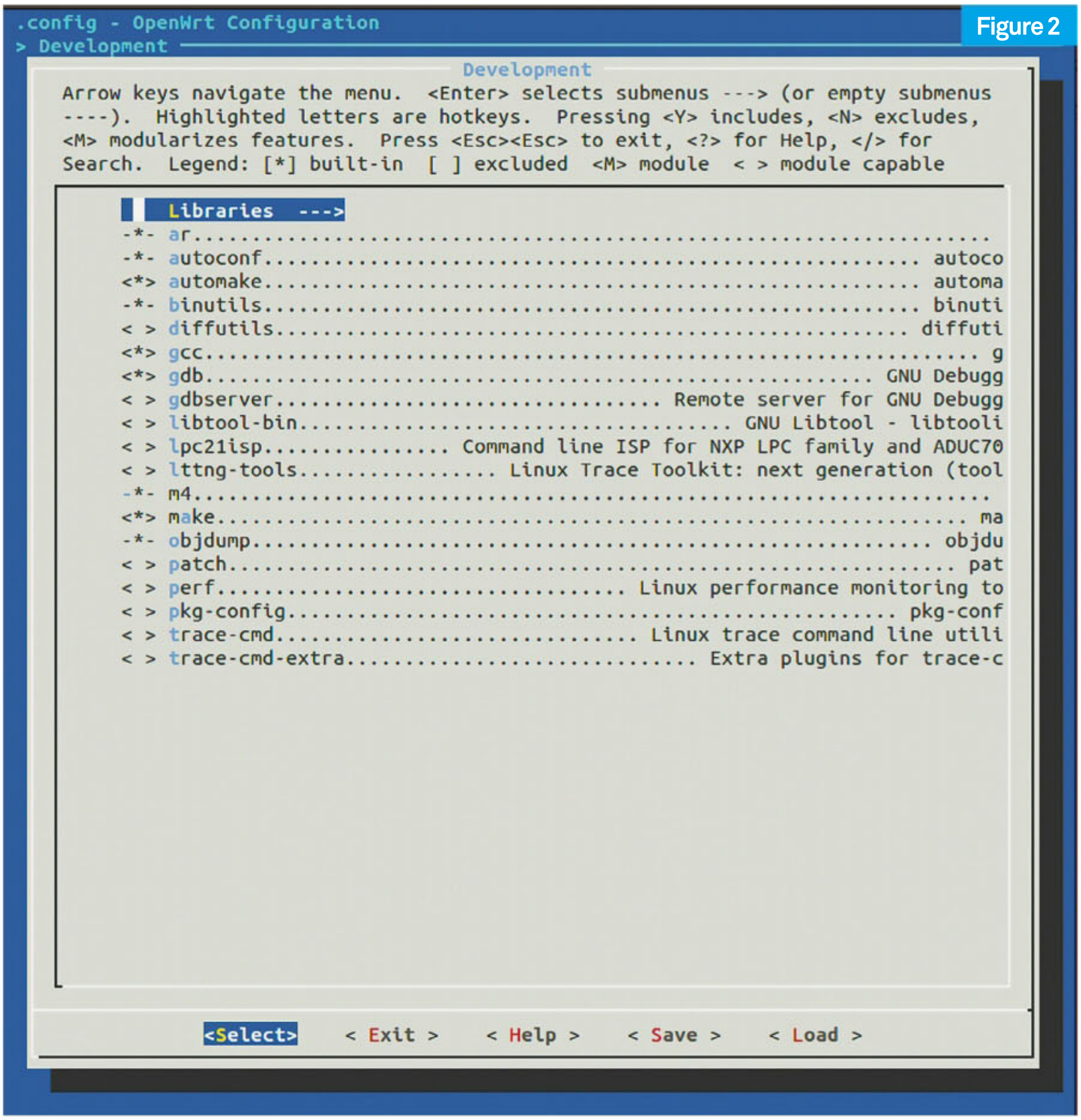


Figure 2

```
.config - OpenWrt Configuration
> Development ─────────────────────────────────────────────────────────────
                              Development
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus
  ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
  <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for
  Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

          Libraries  --->
      -*- ar.......................................................................
      -*- autoconf.......................................................... autoco
      <*> automake.......................................................... automa
      -*- binutils.......................................................... binuti
      < > diffutils........................................................ diffuti
      <*> gcc.................................................................. g
      <*> gdb.......................................................... GNU Debugg
      < > gdbserver................................ Remote server for GNU Debugg
      < > libtool-bin.......................................... GNU Libtool - libtooli
      < > lpc21isp................ Command line ISP for NXP LPC family and ADUC70
      < > lttng-tools................ Linux Trace Toolkit: next generation (tool
      -*- m4.......................................................................
      <*> make.................................................................. ma
      -*- objdump............................................................ objdu
      < > patch............................................................. pat
      < > perf................................. Linux performance monitoring to
      < > pkg-config........................................................ pkg-conf
      < > trace-cmd............................... Linux trace command line utili
      < > trace-cmd-extra............................. Extra plugins for trace-c

             <Select>    < Exit >    < Help >    < Save >    < Load >
```

Right OpenWRT comes with a large selection of developer utilities, which can be deployed on demand previously, however, this is an anomaly: the average user installs OpenWRT on a router which has 4MB, 8MB or 16MB of flash memory.

Either way, move the cursor to the entry for GCC and press **Y** to mark it for deployment in the core image. Due to menuconfig's capability of resolving resulting dependencies, you can see the product mark a few other required packages with a star. In the next step, repeat the steps for GDB and Make to give yourself a realistic chance at compiling.

Should you use the product heavily, marking some of the other utilities might make sense too – your guide's author will select all of them, as this will provoke an educational error. In the next step, use the left and right cursor to move the function highlight to exit and leave the Development section. In the next step, you should also look at the Utilities section, where you can pick a variety of products that you might need.

> “ Look at the Utilities section, where you can pick a variety of products that you might need ”

At that point, it is time to invoke another Make process: due to the extreme size of GCC, a work time of another 30 minutes can be expected. After its completion, an error similar to the one shown in **Figure 3** will pop up.

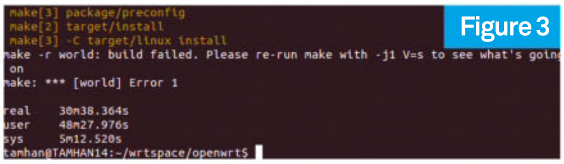


Figure 3

```
 make[3] package/preconfig
 make[2] target/install
 make[3] -C target/linux install
make -r world: build failed. Please re-run make with -j1 V=s to see what's going
 on
make: *** [world] Error 1

real    30m38.364s
user    48m27.976s
sys     5m12.520s
tamhan@TAMHAN14:~/wrtspace/openwrt$
```

Above Uh-oh!

Finding errors in a build is made difficult by a small speciality: the OpenWRT build system, by default, caches any output produced by its lesser tasks. Fortunately, invoking the program with the following parameters can solve this:

```
tamhan@TAMHAN14:~/wrtspace/openwrt$ make -j1
V=s
. . .
Creating filesystem with parameters:
  Size: 50331648
. . .
error: ext4_allocate_best_fit_partial: failed
to allocate 3446 blocks, out of space?
make[5]: *** [mkfs-ext4] Error 1
```

After running for a few minutes, the error will be emitted to the console. In our case, the issue occurs during the creation of the root file system – remember that the image created before was but 78MB in size.

Switching into the Target Images section of menuconfig best solves this problem. There, set the value of 'Root filesystem partition size' to a larger value such as 500MB. After that, rerun the **make** command – it will pass without problem, leading to a new image which can be deployed to your Raspberry Pi.

What‘s the timing?

Your author likes using the **time** command on the compile workstation: simply prepend it to the command that is to be run. It will then output information – the real time value tells you how much clock time has passed, while the other figures provide additional information on resources used. Note that on multicore workstations, the 'lesser' values can be larger than the clock time value if multiple cores are used.

Praise modularity!

Even though GCC can now be invoked, creating custom programs still requires the presence of the proper editor. Recompiling the entire image is a nuisance: let us instead create a package. This process is especially handy when it comes to updating an image of a process computer that is already in active use, since overwriting the image usually destroys all information already found on the router.

This problem is best solved by creating a dedicated package. Return to **make menuconfig** and go to Utilities>Editors. Then, mark the nano package and press the **M** key to declare it a module, as shown in **Figure 4**.

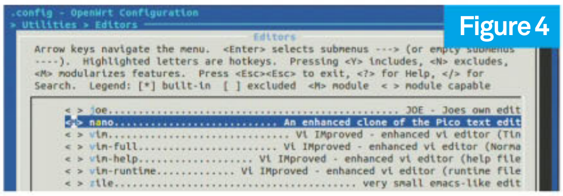


Figure 4

```
.config - OpenWrt Configuration
> Utilities > Editors
                              Editors
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus
  ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
  <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for
  Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

        < > joe........................................... JOE - Joes own edit
        <M> nano........................ An enhanced clone of the Pico text edit
        < > vim.......................... Vi IMproved - enhanced vi editor (Tin
        < > vim-full.................... Vi IMproved - enhanced vi editor (Norma
        < > vim-help.................. Vi IMproved - enhanced vi editor (help file
        < > vim-runtime............. Vi IMproved - enhanced vi editor (runtime file
        < > zile................................... very small emacs-like edit
```

Above This product will be modularised

After saving the configuration, run Make another time to kick off another compile process. When done, a new IPK file will enrich the **packages** folder. Finding it is made easier by using the well-known combination of find and grep:

```
tamhan@TAMHAN14:~/wrtspace/openwrt/bin/
brcm2708/packages$ ls
base  kernel  luci  management  packages
routing  targets  telephony
tamhan@TAMHAN14:~/wrtspace/openwrt/bin/
brcm2708/packages$ find | grep "nano"
./packages/nano_2.7.4-1_brcm2708.ipk
```

This package can then be deployed via the package utility discussed in the previous tutorial (issue 175). Alternatively, the contents of the **packages** folder can also be exposed via a web server, thereby allowing the package manager to obtain relevant packets 'on the fly'.

Conclusion

In the case of the Raspberry Pi, having your own image allows you to add a variety of utilities which the OpenWRT team currently does not provide. Access to GCC means that the world is your oyster – deploying new technologies is as easy as downloading the source code, adapting it to the needs of OpenWRT, compiling it and enjoying the fruits of your labour. ■

Configure a LAMP stack

Paul O'Brien is a professional cross-platform software developer, with extensive experience of deploying and maintaining Linux systems. Android, built on top of Linux, is also one of Paul's specialist topics.

Resources

Ubuntu
ubuntu.com

Apache
apache.org

MySQL
mysql.com

PHP
secure.php.net

Let's Encrypt
letsencrypt.org

certbot-auto
github.com/certbot/certbot

Cloudflare
cloudflare.com

phpMyAdmin
phpmyadmin.net

Webmin
webmin.com

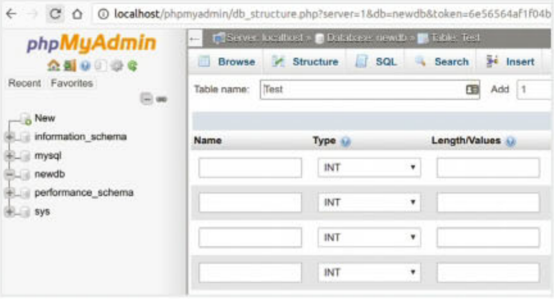
Webmin Authentic Theme
github.com/qooob/authentic-theme

MySQL tuner
mysqltuner.pl

Apache2 tuner
apache2buddy.pl

Percona
percona.com

Deploy your own websites with ease on top of a Linux-Apache-MySQL-PHP stack



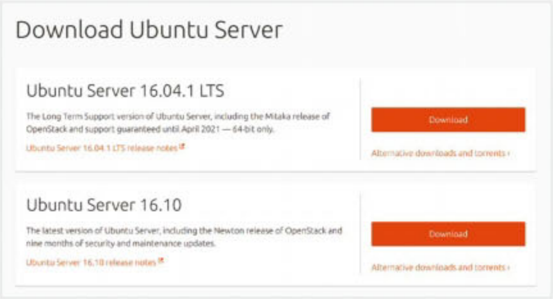
localhost/phpmyadmin/db_structure.php?server=1&db=newdb&token=6e56564af1f04b

Table name: Test Add 1

Name	Type	Length/Values
	INT	
	INT	
	INT	
	INT	

Above phpMyAdmin provides a graphical interface to your database – and configures your LAMP stack as fully operational!

Statistics show that Linux powers around 67 per cent of all web servers, despite having much lower penetration on the desktop. There are lots of reasons for this – for hobbyists the OS is free, for corporations it's powerful yet lightweight (particularly compared to Windows servers) and there is a vast range of software that's ready to deploy on the OS. The most common use case for Linux web servers is a LAMP stack – meaning Linux (OS), Apache (web server), MySQL (database) and PHP (scripting language). With just these four elements installed, it's easy to get many of the most popular websites up and running (such as WordPress) and should you hit any issues, you can tap into a vast library of peer-to-peer support. The stack is well supported on all major Linux distributions and is frequently updated with new features as well as security fixes. No matter how big or small your web project, it is a great place to start.



Download Ubuntu Server

Ubuntu Server 16.04.1 LTS

The Long Term Support version of Ubuntu Server, including the Mitaka release of OpenStack and support guaranteed until April 2021 — 64-bit only.

Ubuntu Server 16.04.1 LTS release notes

Ubuntu Server 16.10

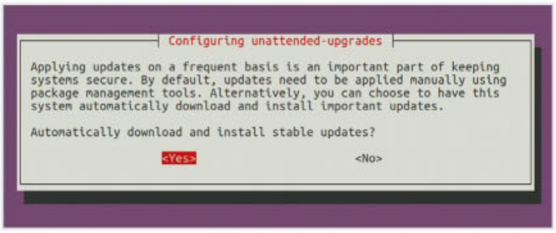
The latest version of Ubuntu Server, including the Newton release of OpenStack and nine months of security and maintenance updates.

Ubuntu Server 16.10 release notes

Download
Alternative downloads and torrents ›
Download
Alternative downloads and torrents ›

01 Choose your distribution

When choosing a distribution for your LAMP server, first think about where it's going to be deployed. If you're going to be using it as your development machine too, then you'll just need to select your preferred desktop distribution. But if you are deploying to a machine in the cloud, a headless machine at home/in the office or a server in a VM such as VirtualBox, then you are best choosing a server-optimised distro. Debian or Ubuntu Server are widely used, frequently updated and highly recommended.



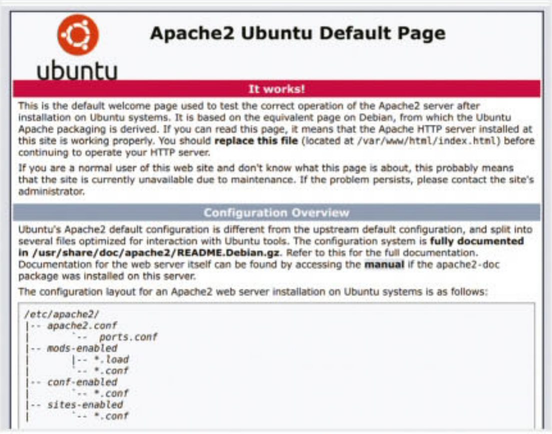
```
Configuring unattended-upgrades

Applying updates on a frequent basis is an important part of keeping
systems secure. By default, updates need to be applied manually using
package management tools. Alternatively, you can choose to have this
system automatically download and install important updates.

Automatically download and install stable updates?

          <Yes>                                   <No>
```

02 Updating and staying up to date

It goes without saying that keeping your server up to date with new releases and security patches is vital, even more so when you are deploying to the cloud. ArubaCloud is one of our favourite providers with decent boxes for only €1 per month, but you'll need to update the distro (we've used Ubuntu Server) after install as the images can be out of date. For automated security updates, simply install the unattended-upgrades package with:

```
sudo dpkg-reconfigure --priority=low unattended-upgrades
```

Even with this enabled, remember to keep an eye on your box!



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

03 Download and install Apache

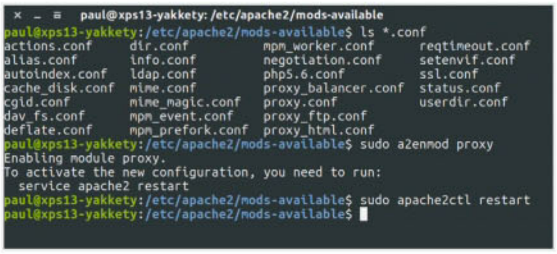
Now that you have got your server deployed and updated, the first step is to install Apache (version 2). All of the elements of the LAMP stack are long established, so as you'd expect, they're available in all major distribution repositories (and frequently updated). Typically there's no need to build from source at all. Simply use :

```
sudo apt-get install apache2
```

...and Apache will be automatically installed and started. The

> Statistics show that Linux powers around 67 per cent of all web servers, despite having much lower penetration on the desktop

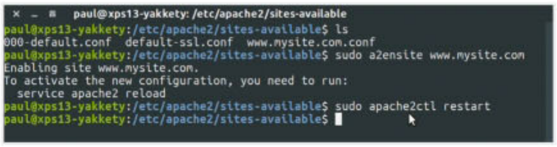
default website will be served on your machine IP – the default files are located at **/var/www**.



```
paul@xps13-yakkety: /etc/apache2/mods-available
paul@xps13-yakkety:/etc/apache2/mods-available$ ls *.conf
actions.conf     dir.conf           mpm_worker.conf       reqtimeout.conf
alias.conf       info.conf          negotiation.conf      setenvif.conf
autoindex.conf   ldap.conf          php5.6.conf           ssl.conf
cache_disk.conf  mime.conf          proxy_balancer.conf   status.conf
cgid.conf        mime_magic.conf    proxy.conf            userdir.conf
dav_fs.conf      mpm_event.conf     proxy_ftp.conf
deflate.conf     mpm_prefork.conf   proxy_html.conf
paul@xps13-yakkety:/etc/apache2/mods-available$ sudo a2enmod proxy
Enabling module proxy.
To activate the new configuration, you need to run:
  service apache2 restart
paul@xps13-yakkety:/etc/apache2/mods-available$ sudo apache2ctl restart
paul@xps13-yakkety:/etc/apache2/mods-available$ 
```

04 Enable modules

After installation, Apache will be available with a default set of modules. Depending on what you plan to use the server for, you may need to enable additional modules (such as rewrite, proxy, SSL etc). To see which Apache modules are available, list the contents of **/etc/apache2/mods-available**. To see which modules are enabled, check **/etc/apache/mods-enabled**. To enable a module, use **a2enmod name**. Typically the defaults will be fine for most standard websites.



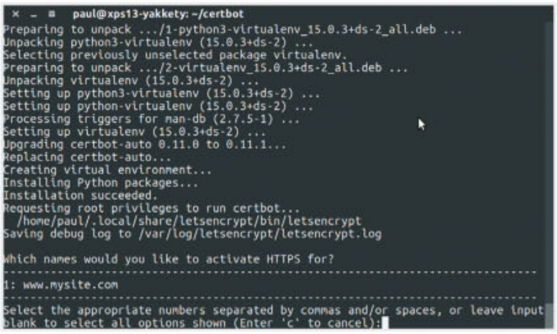
```
paul@xps13-yakkety: /etc/apache2/sites-available
paul@xps13-yakkety:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf  www.mysite.com.conf
paul@xps13-yakkety:/etc/apache2/sites-available$ sudo a2ensite www.mysite.com
Enabling site www.mysite.com.
To activate the new configuration, you need to run:
  service apache2 reload
paul@xps13-yakkety:/etc/apache2/sites-available$ sudo apache2ctl restart
paul@xps13-yakkety:/etc/apache2/sites-available$ 
```

05 Set up individual websites

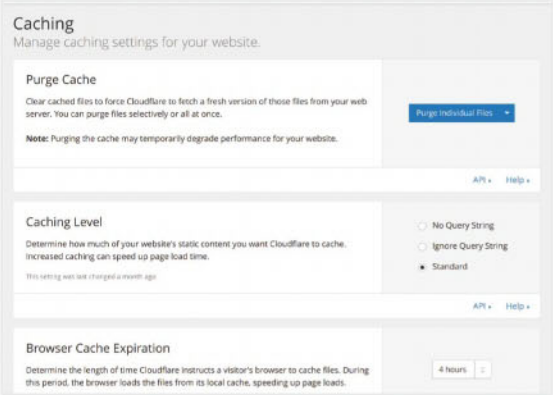
After Apache installation, a single default site will be enabled. Site configuration files live in **/etc/apache2/sites-available**. You may want your Apache install to serve multiple sites, so you'll need to create additional configuration files defining the 'ServerName' you wish to handle and the directory you want to serve files from. After creating the .conf file (eg **www.mysite.com.conf**), you can enable it using the **a2ensite www.mysite.com** command. Remember to restart with **apache2ctl restart** to see changes.

06 Add SSL to your server

After you have your site up and running over regular HTTP, you will likely want to provide access via SSL too. This isn't just for security reasons – Google now actively penalises sites in search results that aren't available using HTTPS, so it's something you definitely want to consider. The good news is that you can get certificates for free using Let's Encrypt from **letsencrypt.org**. Installation is incredibly easy – download the certbot-auto script from **github.com/certbot/certbot**, run it, follow the prompts and you're all set.



```
paul@xps13-yakkety: ~/certbot
Preparing to unpack .../1-python3-virtualenv_15.0.3+ds-2_all.deb ...
Unpacking python3-virtualenv (15.0.3+ds-2) ...
Selecting previously unselected package virtualenv.
Preparing to unpack .../2-virtualenv_15.0.3+ds-2_all.deb ...
Unpacking virtualenv (15.0.3+ds-2) ...
Setting up python3-virtualenv (15.0.3+ds-2) ...
Setting up python-virtualenv (15.0.3+ds-2) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up virtualenv (15.0.3+ds-2) ...
Upgrading certbot-auto 0.11.0 to 0.11.1...
Replacing certbot-auto...
Creating virtual environment...
Installing Python packages...
Installation succeeded.
Requesting root privileges to run certbot...
  /home/paul/.local/share/letsencrypt/bin/letsencrypt
Saving debug log to /var/log/letsencrypt/letsencrypt.log

Which names would you like to activate HTTPS for?
-------------------------------------------------------------------------------
1: www.mysite.com
-------------------------------------------------------------------------------
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
```



Caching

Manage caching settings for your website.

Purge Cache

Clear cached files to force Cloudflare to fetch a fresh version of those files from your web server. You can purge files selectively or all at once.

Note: Purging the cache may temporarily degrade performance for your website.

Caching Level

Determine how much of your website's static content you want Cloudflare to cache. Increased caching can speed up page load time.

This setting was last changed a month ago

○ No Query String
○ Ignore Query String
● Standard

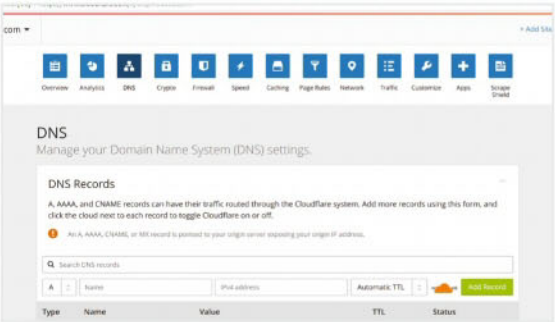
Browser Cache Expiration

Determine the length of time Cloudflare instructs a visitor's browser to cache files. During this period, the browser loads the files from its local cache, speeding up page loads.

4 hours

07 Reduce your bandwidth use with CDN

Use of a CDN (content delivery network) can help reduce the risk of paying excess bandwidth costs from a server provider, as well as improve performance when a site is accessed across the world. Our favourite CDN is Cloudflare (**cloudflare.com**), which will host your static content in its vast network of data centres, reducing load on your main server. The best bit? The standard Cloudflare plan is completely free. Cloudflare is also an alternative way to deploy SSL, should you have any problems with Let's Encrypt.



DNS

Manage your Domain Name System (DNS) settings.

DNS Records

A, AAAA, and CNAME records can have their traffic routed through the Cloudflare system. Add more records using this form, and click the cloud next to each record to toggle Cloudflare on or off.

An A, AAAA, CNAME, or MX record is pointed to your origin server exposing your origin IP address.

08 Find your IP address and manage DNS

In order to test your installation, you need to know the IP address of your server to enter into your browser. On the machine itself, use the command:

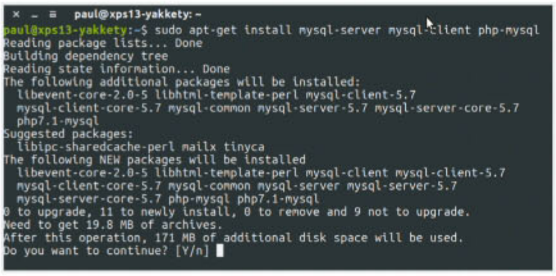
```
ifconfig eth0 | grep inet | awk '{ print $2 }'
```

To avoid having to remember the address every time, you'll likely want to sign up for your own domain name and point a DNS

Percona – optimised MySQL

Percona Server for MySQL is a free, fully compatible, enhanced, open source drop-in replacement for MySQL that provides superior performance, scalability and instrumentation. Downloaded over 3 million times, Percona Server's self–tuning algorithms and support for extremely high-performance hardware delivers excellent performance and reliability. Percona also has an enhanced 'hot backup' facility (named XtraBackup), which is well worth checking out.

entry to the IP address. Again, Cloudflare is a good solution for DNS hosting, with an easy-to-use yet advanced toolkit and an extremely short TTL (time to live), which means DNS changes effectively take immediate effect. If you've ever sat waiting for DNS to propagate, you'll appreciate how valuable this feature is!



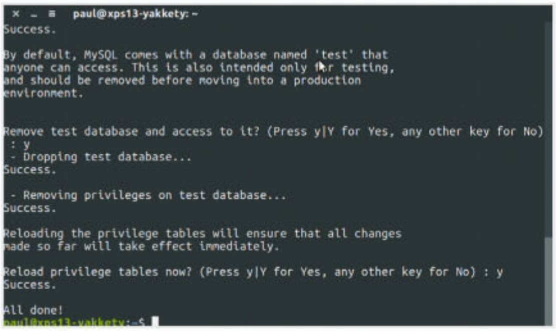
```
paul@xps13-yakkety: ~
paul@xps13-yakkety:~$ sudo apt-get install mysql-server mysql-client php-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libevent-core-2.0-5 libhtml-template-perl mysql-client-5.7
  mysql-client-core-5.7 mysql-common mysql-server-5.7 mysql-server-core-5.7
  php7.1-mysql
Suggested packages:
  libipc-sharedcache-perl mailx tinyca
The following NEW packages will be installed
  libevent-core-2.0-5 libhtml-template-perl mysql-client mysql-client-5.7
  mysql-client-core-5.7 mysql-common mysql-server mysql-server-5.7
  mysql-server-core-5.7 php-mysql php7.1-mysql
0 to upgrade, 11 to newly install, 0 to remove and 9 not to upgrade.
Need to get 19.8 MB of archives.
After this operation, 171 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

09 Download and install MySQL

Now that you have Apache installed, the next step is to install MySQL – the database engine that is used by most Linux web applications. To install, use the command:

```
sudo apt-get install mysql-server mysql-client php-mysql
```

This installs the MySQL server and client (for administration), the Apache module and the PHP module. As part of the install process you'll be prompted to choose a new MySQL root password. Use the **sudo mysql_install_db** command to complete the installation.



```
paul@xps13-yakkety: ~
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No)
 : y
 - Dropping test database...
Success.

 - Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
paul@xps13-yakkety:~$
```

10 Secure your installation

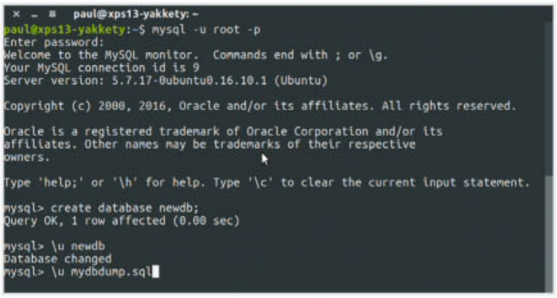
After you've installed (and perhaps tested) MySQL, it's a good idea to run the provided script to secure your installation, using the command:

```
sudo /usr/bin/mysql_secure_installation
```

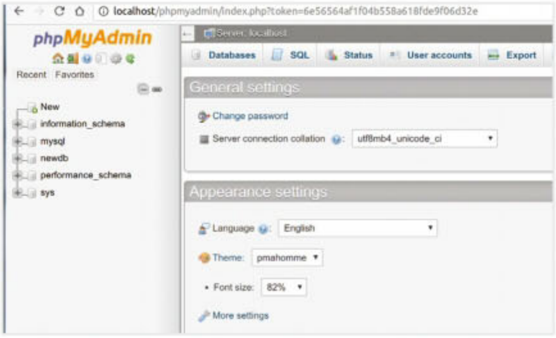
This will remove anonymous users, allow you to change the root password to something more secure if required, allow root access only from localhost and remove the test database. The script will prompt you to reload the privilege tables, which means that the changes will take effect immediately.

11 Administering via the command line

As part of the original MySQL installation, you included 'mysql-client', which provides a command-line client for working with your database. For example: to connect to the MySQL database and create a new database, you'd use the command **mysql -u root -p**, enter your password when prompted, and then use **create database newdbname;** and **\u newdbname**. At this point you could enter SQL statements manually or use **\. filename.sql** to run a script from disk. Remember to set up non-root accounts as required for your applications.



```
paul@xps13-yakkety: ~
paul@xps13-yakkety:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.7.17-0ubuntu0.16.10.1 (Ubuntu)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database newdb;
Query OK, 1 row affected (0.00 sec)

mysql> \u newdb
Database changed
mysql> \u mydbdump.sql
```



localhost/phpmyadmin/index.php?token=6e56564af1f04b558a618fde9f06d32e

New
information_schema
mysql
newdb
performance_schema
sys

Server connection collation: utf8mb4_unicode_ci

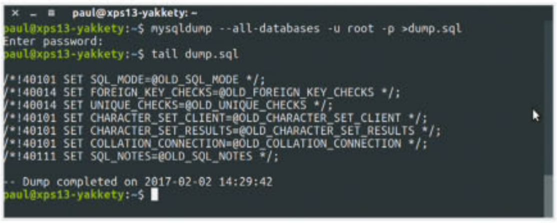
Language: English

Theme: pmahomme

Font size: 82%

12 Use phpMyAdmin for GUI administration

If administering MySQL from the command line feels like a step too far, or you'd just like something a bit more graphical for viewing your data, then you should consider using phpMyAdmin. It's available from the repositories of most distributions; they are often outdated, however, so it's a better idea to download the latest tar.gz from **www.phpmyadmin.net**. Installation is as simple as extracting the archive to a web-accessible directory and setting the key values in the config file.



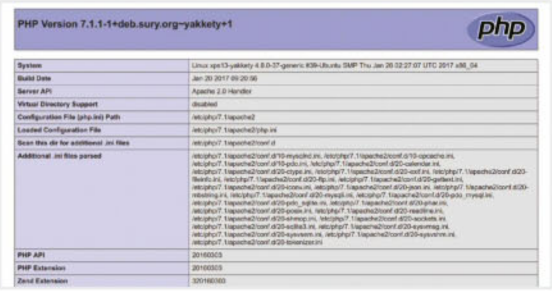
```
paul@xps13-yakkety: ~
paul@xps13-yakkety:~$ mysqldump --all-databases -u root -p >dump.sql
Enter password:
paul@xps13-yakkety:~$ tail dump.sql

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2017-02-02 14:29:42
paul@xps13-yakkety:~$
```

13 Back up your data

You are likely to be storing valuable data on your MySQL server, so you should ensure that you make regular backups. The **mysqldump** command makes it easy to dump the contents of your database to a file, which you can then upload to a remote location (it makes sense to send your **/var/www contents** up, too, for safe keeping). A number of open source automated backup scripts are available as a starting point, which you can then add to Cron to run on a regular basis.

14 Download and install PHP

The final element of the LAMP stack is PHP, the scripting language that powers some of the world's most popular



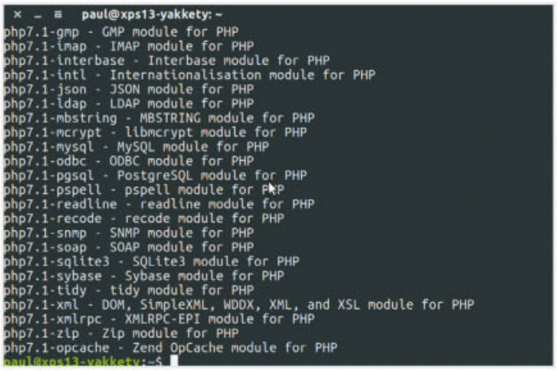
PHP Version 7.1.1-1+deb.sury.org~yakkety+1

System	Linux xps13-yakkety 4.8.0-37-generic #39-Ubuntu SMP Thu Jan 26 02:27:07 UTC 2017 x86_64
Build Date	Jan 20 2017 09:20:56
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.1/apache2
Loaded Configuration File	/etc/php/7.1/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.1/apache2/conf.d
Additional .ini files parsed	/etc/php/7.1/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.1/apache2/conf.d/10-opcache.ini, /etc/php/7.1/apache2/conf.d/10-pdo.ini, /etc/php/7.1/apache2/conf.d/20-calendar.ini, /etc/php/7.1/apache2/conf.d/20-ctype.ini, /etc/php/7.1/apache2/conf.d/20-exif.ini, /etc/php/7.1/apache2/conf.d/20-fileinfo.ini, /etc/php/7.1/apache2/conf.d/20-ftp.ini, /etc/php/7.1/apache2/conf.d/20-gettext.ini, /etc/php/7.1/apache2/conf.d/20-iconv.ini, /etc/php/7.1/apache2/conf.d/20-json.ini, /etc/php/7.1/apache2/conf.d/20-mbstring.ini, /etc/php/7.1/apache2/conf.d/20-mysqli.ini, /etc/php/7.1/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.1/apache2/conf.d/20-pdo_sqlite.ini, /etc/php/7.1/apache2/conf.d/20-phar.ini, /etc/php/7.1/apache2/conf.d/20-posix.ini, /etc/php/7.1/apache2/conf.d/20-readline.ini, /etc/php/7.1/apache2/conf.d/20-shmop.ini, /etc/php/7.1/apache2/conf.d/20-sockets.ini, /etc/php/7.1/apache2/conf.d/20-sqlite3.ini, /etc/php/7.1/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.1/apache2/conf.d/20-sysvsem.ini, /etc/php/7.1/apache2/conf.d/20-sysvshm.ini, /etc/php/7.1/apache2/conf.d/20-tokenizer.ini
PHP API	20160303
PHP Extension	20160303
Zend Extension	320160303

applications. To install PHP, use the command:

```
sudo apt-get install php libapache2-mod-php
```

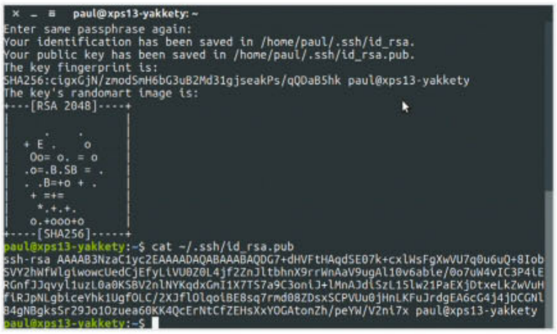
This installs the latest version of PHP together with the Apache module required to integrate (it will be automatically enabled). Note that some older applications explicitly require php5 rather than 7, so you may need to adjust the command accordingly. You can test your PHP install by creating a **test.php** file with the content **<?php phpinfo(); ?>**.



```
paul@xps13-yakkety: ~
php7.1-gmp - GMP module for PHP
php7.1-imap - IMAP module for PHP
php7.1-interbase - Interbase module for PHP
php7.1-intl - Internationalisation module for PHP
php7.1-json - JSON module for PHP
php7.1-ldap - LDAP module for PHP
php7.1-mbstring - MBSTRING module for PHP
php7.1-mcrypt - libmcrypt module for PHP
php7.1-mysql - MySQL module for PHP
php7.1-odbc - ODBC module for PHP
php7.1-pgsql - PostgreSQL module for PHP
php7.1-pspell - pspell module for PHP
php7.1-readline - readline module for PHP
php7.1-recode - recode module for PHP
php7.1-snmp - SNMP module for PHP
php7.1-soap - SOAP module for PHP
php7.1-sqlite3 - SQLite3 module for PHP
php7.1-sybase - Sybase module for PHP
php7.1-tidy - tidy module for PHP
php7.1-xml - DOM, SimpleXML, WDDX, XML, and XSL module for PHP
php7.1-xmlrpc - XMLRPC-EPI module for PHP
php7.1-zip - Zip module for PHP
php7.1-opcache - Zend OpCache module for PHP
paul@xps13-yakkety:~$
```

15 Install PHP modules

Just as with Apache, PHP is extensible with modules. To see which modules are available, use the command **apt-cache search php-**. Applications will typically specify which PHP modules they require; if you are seeing issues with blank pages, however, you may have PHP error reporting turned off, hiding the cause of the problem. You can temporarily enable it in your scripts with the lines **error_reporting(E_ALL);** and **ini_set('display_errors', '1')**. The Apache log in **/var/log/apache2** is also a good place to look for clues.

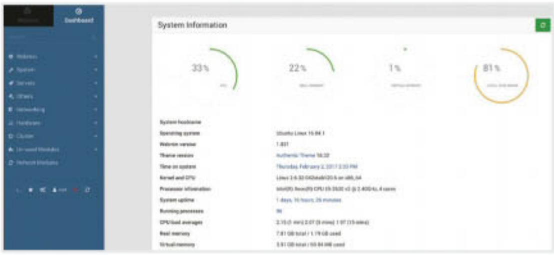


```
paul@xps13-yakkety: ~
Enter same passphrase again:
Your identification has been saved in /home/paul/.ssh/id_rsa.
Your public key has been saved in /home/paul/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:cigxGjN/zmodSmH6bG3uB2Md31gjseakPs/qQDaB5hk paul@xps13-yakkety
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|      .    .     |
|   + E .    o    |
|    Oo= o. = o   |
|   .o=.B.SB = .  |
|   . .B=+o + .   |
|    + =+=        |
|     *.+.+.      |
|    o.+ooo+o     |
+----[SHA256]-----+
paul@xps13-yakkety:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDG7+dHVFtHAqdSE07k+cxlWsFgXwVU7q0u6uQ+8Iob
SVY2hWfWlgiwowcUedCjEfyLiVU0Z0L4jf2ZnJltbhnX9rrWnAaV9ugAl10v6able/0o7uW4vIC3P4iE
RGnfJJqvyl1uzL0a0KSBV2nlNYKqdxGmI1X7TS7a9C3oniJ+lMnAJdiSzL15lw21PaEXjDtxeLkZwVuH
fiRJpNLgbiceYhk1UgfOLC/2XJflOlqoiBE8sq7rmd08ZDsxSCPVUu0jHnLKFuJrdgEA6cG4j4jDCGNl
B4gNBgksSr29Jo1Ozuea60KK4QcErNtCfZEHsXxYOGAtonZh/peYW/V2ni7x paul@xps13-yakkety
paul@xps13-yakkety:~$
```

16 Uploading to your server

If you are deploying your LAMP stack on your local machine (for test purposes perhaps), then uploading content isn't a challenge. For cloud-based or VM deployments, however, this may not be the case. The easiest way to deploy is via SFTP. Ensure you have openssh-server installed on the server. Generate a public key on your machine (**ssh-keygen -t rsa**), add the content of **~/.ssh/id_rsa.pub** to **~/.ssh/authorized_keys** on the server (for the appropriate user) and you should be able to connect directly with the **sftp** command.

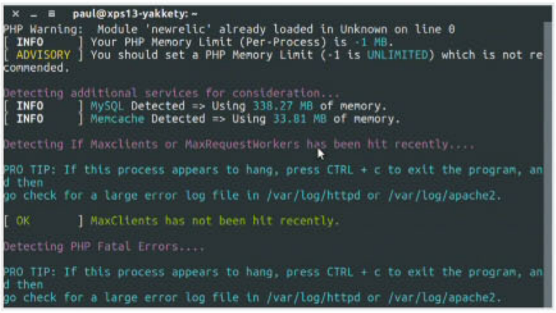
> "You are likely to be storing valuable data on your MySQL server, so you should ensure that you make regular backups"



System hostname	
Operating system	Ubuntu Linux 16.04.1
Webmin version	1.831
Theme version	Authentic Theme 18.32
Time on system	Thursday, February 2, 2017 2:33 PM
Kernel and CPU	[illegible]
Processor information	[illegible]
System uptime	[illegible]
Running processes	[illegible]
CPU load averages	[illegible]
Real memory	7.81 GB total / 1.19 GB used
Virtual memory	[illegible]

17 GUI-based server admin

For GUI-based server admin, one of our favourite tools is Webmin. Available in an apt repository from **www.webmin.com/deb.html**, once installed, Webmin provides a graphical interface to all aspects of your Linux system, including all elements of the LAMP stack. The stock Webmin theme is very outdated, but the third-party Authentic theme brings Webmin straight into the 21st century with a Bootstrap-based, responsive theme. Grab it from **github.com/qooob/authentic-theme**.



```
paul@xps13-yakkety: ~
PHP Warning:  Module 'newrelic' already loaded in Unknown on line 0
[ INFO     ] Your PHP Memory Limit (Per-Process) is -1 MB.
[ ADVISORY ] You should set a PHP Memory Limit (-1 is UNLIMITED) which is not re
commended.

Detecting additional services for consideration...
[ INFO     ] MySQL Detected => Using 338.27 MB of memory.
[ INFO     ] Memcache Detected => Using 33.81 MB of memory.

Detecting If Maxclients or MaxRequestWorkers has been hit recently....

PRO TIP: If this process appears to hang, press CTRL + c to exit the program, an
d then
go check for a large error log file in /var/log/httpd or /var/log/apache2.

[ OK       ] MaxClients has not been hit recently.

Detecting PHP Fatal Errors....

PRO TIP: If this process appears to hang, press CTRL + c to exit the program, an
d then
go check for a large error log file in /var/log/httpd or /var/log/apache2.
```

18 Tune your installation

So that's it, you're all set up! Out of the box, Apache and MySQL use a standard set of config files that work fine on most hardware but are certainly not optimised for your specific setup. Two Perl scripts, available from **mysqltuner.pl** and **apache2tuner.pl**, are designed to examine all the configuration files together with your system config and suggest improvements based on your system. They also look at logs from your running system, so ensure everything is up for at least 24 hours before use.

Securing your installation

There are simple steps you can take to help secure your LAMP environment. For example, you can disable directory listing on your server by including the **<Directory>Options -Indexes</Directory>** directive in your Apache config. You should also set the ServerTokens Prod and ServerSignature Off directives to avoid potential hackers learning about your server version/configuration. In the same way, add **expose_php = Off** in your PHP config to keep its version hidden. It's a good idea to disable unused Apache and PHP modules too, and you can disable potentially dangerous PHP functions with:

```
disable_functions
=exec,shell_exec,
passthru,system,
popen,curl_exec,
curl_multiexec,
parse_ini_file,
show_source,
proc_open,
pcntl_exec
```

...in your config.

Sandbox applications with Firejail

Run a potentially insecure application in a sandbox to protect your system and data

Nitish Tiwari
is a software developer by profession, with a huge interest in free and open source software. As well as serving as community moderator and author for leading FOSS publications, he helps organisations adopt open source software for their business needs.

Resources

Firejail
firejail.wordpress.com

The internet is the biggest source of knowledge out there and is widely used to share information, but it is also the biggest source of malware, spyware and several other types of viruses. This makes the internet a very interesting, yet difficult place to be in. You need to be careful of unverified (and sometimes even verified) software that you download from the internet, the sites you visit, the email attachments you open and so on. This can be difficult to do for anyone, whether they are from a technical background or not.

While nothing can replace being alert to the dangers to system security, there are tools that can help you sandbox the applications you run. This way you can be sure that an application downloaded from the internet does just what it is supposed to do. Here, we will introduce you to Firejail. It is a simple tool to help you restrict the running environment of untrusted applications using Linux namespaces.

Firejail allows a process and all its descendants to have their own private view of the globally shared kernel resources, such as the network stack, process table and mount table. Firejail is written in C with virtually no dependencies, and the software runs on any Linux computer with a 3.x kernel version or newer. It can sandbox any type of process: servers, graphical applications and even user login sessions. Please note that we have taken Ubuntu 16.04 as our host system for this tutorial.

01 Installation

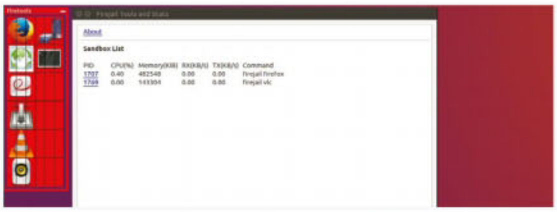
Prebuilt Firejail packages are available for all the popular OS distributions – like Debian, Ubuntu, Linux Mint, Fedora, openSUSE, CentOS 7 and RHEL 7 – from its website. Note that these packages require a 64-bit system. Firejail is available in the AUR for Arch Linux and you can find packages for Slackware from the SlackBuilds repository. To install Firejail on Ubuntu, execute this command in a terminal:

```
$ sudo apt-get install firejail
```

If prompted, type **Y** for installation to continue. Now you are ready to jail your applications using Firejail.

You can also install Firejail from the source code. First, download the archive and extract the files with:

```
$ tar -xjvf firejail-X.Y.Z.tar.bz2
```



Sandbox List

PID	CPU(%)	Memory(KiB)	RX(KB/s)	TX(KB/s)	Command
1707	0.40	482548	0.00	0.00	firejail firefox
1709	0.00	143304	0.00	0.00	firejail vlc

Right Applications sandboxed in Firejail; the window on the right side displays the processes running in a sandbox

Or using:

```
$ tar -xJvf firejail-X.Y.Z.tar.xz
```

Then compile and install:

```
$ cd firejail-X.Y.Z
$ ./configure && make && sudo make install-strip
```

To install in Debian and Mint, download the DEB package and install it with:

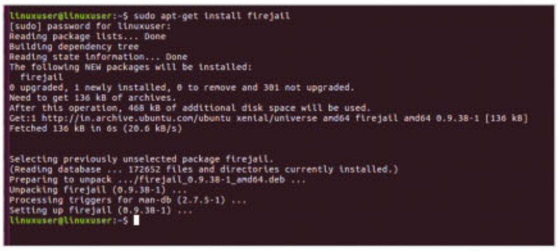
```
$ sudo dpkg -i firejail_X.Y_1_amd64.deb
```

Or using:

```
$ sudo dpkg -i firejail_X.Y_1_i386.deb
```

For Fedora, openSUSE and CentOS 7, download the RPM package and install it:

```
$ sudo rpm -i firejail_X.Y-Z.x86_64.rpm
```



```
linuxuser@linuxuser:~$ sudo apt-get install firejail
[sudo] password for linuxuser:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  firejail
0 upgraded, 1 newly installed, 0 to remove and 301 not upgraded.
Need to get 136 kB of archives.
After this operation, 468 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu xenial/universe amd64 firejail amd64 0.9.38-1 [136 kB]
Fetched 136 kB in 6s (20.6 kB/s)


Selecting previously unselected package firejail.
(Reading database ... 172652 files and directories currently installed.)
Preparing to unpack .../firejail_0.9.38-1_amd64.deb ...
Unpacking firejail (0.9.38-1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up firejail (0.9.38-1) ...
linuxuser@linuxuser:~$ 
```

02 Using Firejail

All types of programs – GUI or CLI based – work well with Firejail. So, you can be confident while opening any untrusted program. To jail a program in the system, start with prefixing 'firejail' to it. The general command format is:

```
$ firejail [options] program_and_arguments
```

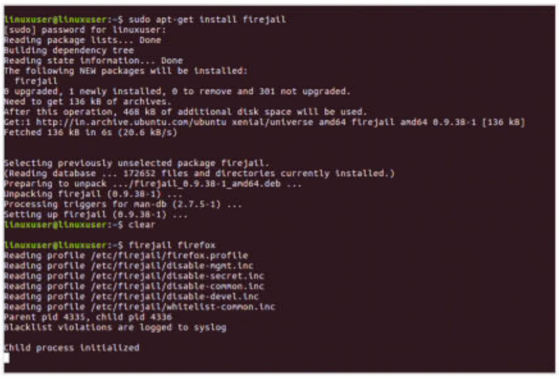
For example, if you want to run Firefox, execute:

```
$ firejail firefox
```

If you want to hide all the files in your home directory from sandboxed programs, execute the firejail command in private mode:

```
$ firejail --private program_and_arguments
```

Until the sandbox runs, Firejail mounts a temporary file system on top of the **/home/user** directory. Any files created in this directory will be deleted once the sandbox



```
linuxuser@linuxuser:~$ sudo apt-get install firejail
[sudo] password for linuxuser:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  firejail
0 upgraded, 1 newly installed, 0 to remove and 301 not upgraded.
Need to get 136 kB of archives.
After this operation, 468 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu xenial/universe amd64 firejail amd64 0.9.38-1 [136 kB]
Fetched 136 kB in 6s (20.6 kB/s)


Selecting previously unselected package firejail.
(Reading database ... 172652 files and directories currently installed.)
Preparing to unpack .../firejail_0.9.38-1_amd64.deb ...
Unpacking firejail (0.9.38-1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up firejail (0.9.38-1) ...
linuxuser@linuxuser:~$ clear

linuxuser@linuxuser:~$ firejail firefox
Reading profile /etc/firejail/firefox.profile
Reading profile /etc/firejail/disable-mgmt.inc
Reading profile /etc/firejail/disable-secret.inc
Reading profile /etc/firejail/disable-common.inc
Reading profile /etc/firejail/disable-devel.inc
Reading profile /etc/firejail/whitelist-common.inc
Parent pid 4335, child pid 4336
Blacklist violations are logged to syslog

Child process initialized
```

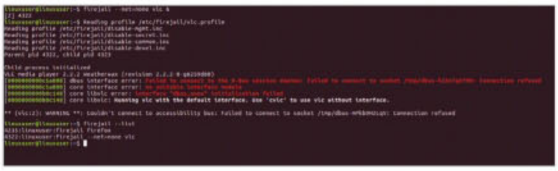
exits. To use the existing directory as the home directory for your sandbox instead and have a persistent sandbox on home, execute:

```
$ firejail /etc/init.d/apache2 start
```

To start servers in Firejail you will need root permission for it. To start Apache server, execute:

```
$ firejail /etc/init.d/apache2 start
```

Note that if you execute Firejail without any arguments, it starts the regular **/bin/bash** shell. Only the Bash session and its descendants are visible inside the sandbox.



```
linuxuser@linuxuser:~$ firejail --net=none vlc &
[2] 4322
linuxuser@linuxuser:~$ Reading profile /etc/firejail/vlc.profile
Reading profile /etc/firejail/disable-mgmt.inc
Reading profile /etc/firejail/disable-secret.inc
Reading profile /etc/firejail/disable-common.inc
Reading profile /etc/firejail/disable-devel.inc
Parent pid 4322, child pid 4323

Child process initialized
VLC media player 2.2.2 Weatherwax (revision 2.2.2-0-g6259d80)
[illegible] dbus interface error: Failed to connect to the D-Bus session daemon: [illegible] Connection refused
[illegible] core interface error: no suitable interface module
[illegible] core libvlc error: interface "dbus,none" initialization failed
[illegible] core libvlc: Running vlc with the default interface. Use 'cvlc' to use vlc without interface.

** (vlc:2): WARNING **: Couldn't connect to accessibility bus: Failed to connect to socket /tmp/dbus-[illegible]: Connection refused
linuxuser@linuxuser:~$ firejail --list
4335:linuxuser:firejail firefox
4322:linuxuser:firejail --net=none vlc
linuxuser@linuxuser:~$
```

03 Using Firejail, continued

To view the list of running sandboxes, use the following command:

```
$ firejail --list
```

You will see the list in the format PID:username:command. To view the process tree within each sandbox, type:

```
$ firejail --tree
```

If you want to join an already running sandbox, obtain the PID of the sandbox using the **--list** option. Then pass this PID into the following command:

> “If you execute Firejail without any arguments, it starts the regular /bin/bash shell. Only the Bash session and its descendants are visible inside the sandbox”

```
$ firejail --join=3974
```

If you are starting Firefox in Firejail and you want a browser-specific iptables filter, execute:

```
$ firejail --net=eth0 firefox
```

This creates a new TCP/IP stack for the Firefox session, assigns an IP address, and installs a browser-specific iptables filter. If the program you are starting in Firejail doesn't need network access, execute:

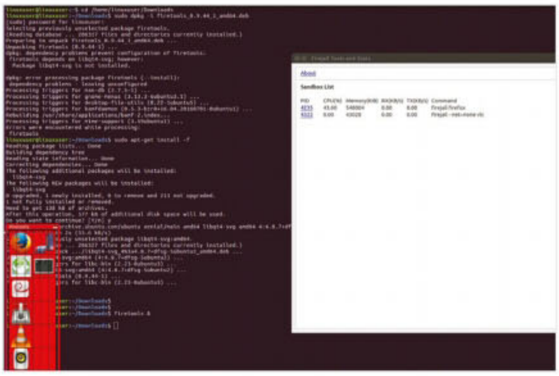
```
$ firejail --net=none vlc
```

Control the amount of data that flows into and out of sandboxes:

```
$ firejail --name=browser --net=eth0 firefox &
$ firejail --bandwidth=browser set eth0 70 30
```

This example sets a bandwidth of 70 kilobytes per second on the receive side and a bandwidth of 30 kilobytes per second on the transmit side. You can change this at runtime and clear them as well, using:

```
$ firejail --bandwidth=browser clear eth0
```



04 Firetools

Firetools is the GUI for Firejail. It is distributed as a separate package and is built using Qt4/Qt5 libraries. It has a sandbox launcher option with the system tray, sandbox editing, management and statistics. It isn't in standard repositories, but an official DEB is available. Download the relevant Firetools DEB file, then navigate to the folder where the file is downloaded and execute:

```
$ sudo dpkg -i firetools*.deb
```

This installs Firetools. However, in case of a dependency issue, resolve it using:

```
$ sudo apt-get install -f
```

Start it using:

```
$ firetools &
```

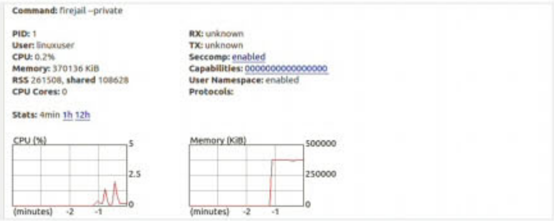
Network setup for sandbox applications

Network connection to a sandbox application can be direct or routed. In a direct setup, a sandbox is connected to the network using the kernel MAC VLAN device; in a routed setup, it uses the bridge device. A direct setup looks like this:

```
$ firejail
-- net=eth0
firefox
```

In a routed setup, first set up the bridge and then pass the bridge name with the **--net** parameter in the above command.

You can see the Firetools window and an indicator for running the app in the system tray. To start any app, double-click it, or right-click and select Run. To monitor the running apps, right-click anywhere in the GUI and select Tools; all the running apps will be listed. To add a new app in Firetools, right-click the GUI and click Edit. Fill in the relevant details with the Command section value as the Firejail app name.



Command: firejail --private

PID: 1
User: linuxuser
CPU: 0.2%
Memory: 370136 KiB
RSS 261508, **shared** 108628
CPU Cores: 0

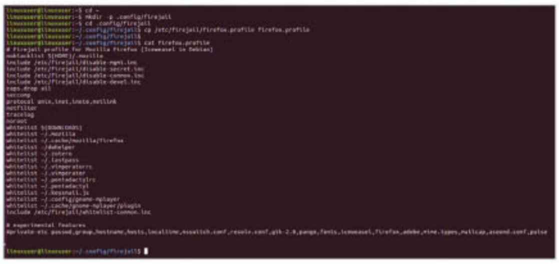
RX: unknown
TX: unknown
Seccomp: enabled
Capabilities: 0000000000000000
User Namespace: enabled
Protocols:

Stats: 4min 1h 12h

05 Firejail features

The core technology behind Firejail is Linux namespaces. This lightweight visualisation technology is used as the first step of isolating the application.

The application container is built automatically when the sandbox starts and is destroyed when the sandbox is closed. The container is based on the file system currently installed. Firejail can attach a new TCP/IP networking stack to the sandbox, which comes with its own routing table, firewall and set of interfaces. This stack is independent of the host network stack. Networking operations supported by Firejail are: create new interfaces, move existing interfaces, assign addresses, hostname support, DNS support, Linux netfilter support and traffic shaping. The security profile file is located in the **/etc/firejail** directory and it describes the file system container, the security filters and network configuration. AppImage in Firejail does native format packaging. Add **--appimage** to mount the package and run it inside the sandbox. Firejail provides sandbox auditing and monitoring. It can point out gaps in security profiles. It provides options to track all the aspects of sandboxed applications. Monitoring of CPU/memory/bandwidth usage, tracing system calls, monitoring exec and fork events can be done using Firetools.



```
linuxuser@linuxuser:~$ cd ~
linuxuser@linuxuser:~$ mkdir -p .config/firejail
linuxuser@linuxuser:~$ cd .config/firejail
linuxuser@linuxuser:~/.config/firejail$ cp /etc/firejail/firefox.profile firefox.profile
linuxuser@linuxuser:~/.config/firejail$
linuxuser@linuxuser:~/.config/firejail$ cat firefox.profile
# Firejail profile for Mozilla Firefox (Iceweasel in Debian)
noblacklist ${HOME}/.mozilla
include /etc/firejail/disable-mgmt.inc
include /etc/firejail/disable-secret.inc
include /etc/firejail/disable-common.inc
include /etc/firejail/disable-devel.inc
caps.drop all
seccomp
protocol unix,inet,inet6,netlink
netfilter
tracelog
noroot
whitelist ${DOWNLOADS}
whitelist ~/.mozilla
whitelist ~/.cache/mozilla/firefox
whitelist ~/dwhelper
whitelist ~/.zotero
whitelist ~/.lastpass
whitelist ~/.vimperatorrc
whitelist ~/.vimperator
whitelist ~/.pentadactylrc
whitelist ~/.pentadactyl
whitelist ~/.keysnail.js
whitelist ~/.config/gnome-mplayer
whitelist ~/.cache/gnome-mplayer/plugin
include /etc/firejail/whitelist-common.inc

# experimental features
#private-etc passwd,group,hostname,hosts,localtime,nsswitch.conf,resolv.conf,gtk-2.0,pango,fonts,iceweasel,firefox,adobe,mime.types,mailcap,asound.conf,pulse

linuxuser@linuxuser:~/.config/firejail$
```

06 Building custom profiles

All the custom profiles are stored within the **~/.config/firejail** directory. Pass Firejail command-line configuration options to the program using profile files.

To build a custom profile for any app, first create the **.config/firejail** directory in your home directory:

```
$ cd ~
$ mkdir -p .config/firejail
$ cd .config/firejail
```

Copy the default security profiles used by Firejail into this directory using:

```
$ cp /etc/firejail/app_name.profile app_name.
profile
```

Here, **app_name** is the name of the app that you're creating a custom profile for. Now edit the new profile. You can comment out lines, blacklist directories, whitelist files, etc. Once modified, save and close the file, then start your application using:

```
$ firejail app_name
```

As a profile file with the same name as the application is present in the **~/.config/firejail** directory, it will be loaded. You can also create a profile file in the **/etc/firejail** directory. If the profile file with app name is present in both the directories, **~/.config/firejail** takes precedence over **/etc/firejail**. A profile file can also be created at a different location and specified while starting the application.

07 Building whitelist profiles

Whitelist profiles are more restrictive profiles and are built using Firejail's whitelisting feature. The custom profiles we created in the last section are also called blacklisted profiles, in that you blacklist the files the application is not allowed to use. Here, files that are necessary for the application to run are listed while everything else is not accessible. To create a whitelist profile, first create a simple Bash sandbox using **--private**:

```
$ firejail --private
```

Now start the app in this Bash session and list all the files within it using:

```
$ find .
```

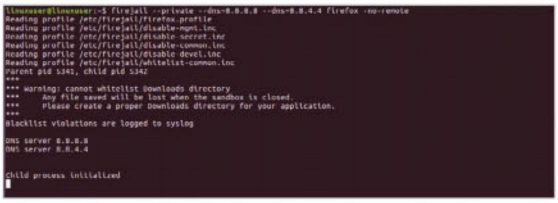
Locate the directory/file that needs to be whitelisted, then exit and close the sandbox. Then, as before, create a new profile in the **~/.config/firejail** directory with the name **app_name.profile**. Enter the following as the content of the file:

```
$ app_name profile
mkdir ~/. directory_name
whitelist ~/.directory_name
include /etc/firejail/whitelist-common.inc
include /etc/firejail/default.profile
```

Here, we are whitelisting **directory_name**, which is within the **app_name** directory structure. Session configurations are included from **whitelist-common.inc**.

08 Firefox sandboxing

Start a Firefox process by executing:



```
linuxuser@linuxuser:~$ firejail --private --dns=8.8.8.8 --dns=8.8.4.4 firefox -no-remote
Reading profile /etc/firejail/firefox.profile
Reading profile /etc/firejail/disable-mgmt.inc
Reading profile /etc/firejail/disable-secret.inc
Reading profile /etc/firejail/disable-common.inc
Reading profile /etc/firejail/disable-devel.inc
Reading profile /etc/firejail/whitelist-common.inc
Parent pid 5341, child pid 5342
***
*** Warning: cannot whitelist Downloads directory
***     Any file saved will be lost when the sandbox is closed.
***     Please create a proper Downloads directory for your application.
***
Blacklist violations are logged to syslog

DNS server 8.8.8.8
DNS server 8.8.4.4


Child process initialized
```

```
$ firejail firefox
```

A single instance of Firefox can have multiple browser windows. So, if Firefox is running already and you want to start a new session within Firejail, execute:

```
$ firejail firefox -no-remote
```

Firefox is allowed only a small set of files and directories by the sandbox. Log messages are sent to syslog if Firefox tries to access blacklisted files. The security filters enabled by default reduce the attack surface of the kernel. The seccomp filter is generally used by default. If you don't trust the add-ons and plug-ins installed in your browser, use the **--private** option and configure the DNS at sandbox level. This prevents attacks from reconfiguring DNS.

```
$ firejail --private --dns=8.8.8.8 --dns=8.8.4.4
firefox -no-remote
```

To create a new TCP/IP stack, connect to your Ethernet network and give an explicit IP address:

```
$ firejail --net=eth0 --ip=192.168.1.207 firefox
```

Firejail replaces the regular X11 server with an Xpra or Xephyr server, preventing X11 keyboard loggers and screenshot utilities from accessing the main X11 server.

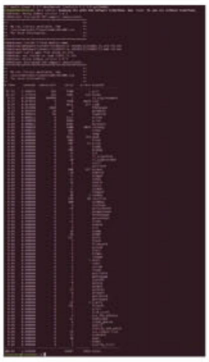
```
$ firejail --x11 --net=eth0 firefox
```

09 Seccomp filter

Firejail uses Seccomp-bpf by default. It stands for 'secure computing mode', which is expressed in Berkeley Packet Filter (BPF) format. Build a whitelist seccomp filter and attach it to a user program using the Firejail sandbox. Firejail needs the syscalls details for seccomp filter. We used strace for getting syscalls. Let's take VLC media player as an example:

```
$ strace -qcf vlc
```

This opens VLC; once you close it, strace prints the syscall list. Copy the list of all syscalls into a text editor and make them comma-separated without any space in between. Now use this comma-separated syscall string in **--seccomp**. Keep the values below:



```
$ firejail --shell=none --seccomp.
keep=poll,futex,[...] vlc
```

The command looks lengthy, but make sure all the syscalls are included. In case of error, add the missed syscall. Now create a whitelist profile in ~/.config/firejail. Copy the default VLC profile here and add the two lines below to it:

```
shell none
seccomp.keep poll,futex,rt_
sigtimedwait,stat,read,[...]
```

Note that the dots denote the rest of the comma-separated syscall names.

10 AppImage support

AppImage is a universal software packaging format developed by Simon Peter.

Start your AppImage application in Firejail using:

```
$ firejail --appimage=Firefox-
50.1.0.glibc2.3.3-x86_64.AppImage
```

All sandbox options are available, like **--private** or **--x11**. Suppose you have downloaded Firefox from the AppImage project repository; to start the sandbox, use:

```
$ firejail --appimage --private --net=eth0 --x11
~/Downloads/Firefox-50.1.0.glibc2.3.3-x86_64.AppImage
```

--appimage is used to enter AppImage mode, **--private** creates an empty home directory, **-net=eth0** creates new network namespace and **--x11** is used for Xpra-based sandboxing.

To verify the security parameters, open Firetools and click Firefox sandbox to get the stats. See that seccomp status that is enabled and the capability field that is zero? These two parameters are most important for a sandbox and everything else is built on top of them. If you have more than one sandbox running, limit the bandwidth of each sandbox as follows:

```
$ firejail --bandwidth=32119 set eth0 80 20
```

> Firejail can attach a new TCP/IP networking stack to the sandbox, which comes with its own routing table, firewall and set of interfaces. This stack is independent of the host network stack

Firejail sandboxing for X11

Xpra and Xephyr are supported in Firejail X11 sandboxing. Regular X11 is replaced by these X11 software packages:

```
$ firejail
-- x11=xpra
-- net=eth0
program-and-
arguments
```

```
$ firejail
-- x11=xephyr
-- net=eth0
program-and-
arguments
```

Another option is to use:

```
$ firejail
-- x11 --
net=eth0
program-and-
arguments
```

This way, Firejail first tries Xpra and if not installed, it tries to find Xephyr. Note that if a sandbox is started as the root user, X11 sandboxing features are not supported.

Mihalis Tsoukalos
is a UNIX administrator, a programmer (UNIX and iOS), a DBA and a mathematician. He has been using Linux since 1993. You can reach him at **@mactsouk** (Twitter) and his website: **mtsoukalos.eu**

Resources

An installation of Erlang

A text editor such as Emacs or vi

Tutorial files available:

filesilo.co.uk

Program in Erlang: File I/O

Discover how to work with file operations in Erlang

No programming language can be considered important if it cannot deal with text and binary files, because you can't develop practical system tools without reading data from existing files or storing your data to files. As a result, the subject of this tutorial will be the use of Erlang to process files and directories, which also includes Erlang functions that allow you to find information about files.

The next Erlang tutorial will talk about network programming in Erlang, which is also a very significant and essential task, so keep reading and write as much Erlang code as possible.

About file I/O

File operations are a critical part of systems programming and as such they must be reliable and fast; however, in this case reliable is more important than fast! A large part of reliability comes from error-checking and error-reporting code, because you cannot be sure what might fail and why.

Most of the functions related to file operations belong to the file module that gives you an interface to the file system. The two most important Erlang functions of the file module that are related to file input and output are file:read() and file:write().

The first one is used for reading from an open file whereas the second is used for writing to a file. The file:read() function takes two arguments: an open file descriptor and the number of bytes you want to read. The file:write() function also requires two arguments: the open file descriptor of the file you want to write to and the data you want to write, respectively. As you will see, there are easier ways to read a file.

You will also need two more functions in order to open and close a file, which are open(File, Mode) and close(Device) respectively. The file:open() function takes two arguments, which are the location of the file you want to open and the mode in which you want to open the file. A successful call to file:open() returns a tuple of type {ok, Fd}. The first element shows that the operation was successful and the second element is used for referencing the open file afterwards. The Mode part can have four values: Read, Write, Append and Exclusive.

Erlang offers the file:read_file() function that takes one argument, which is a filename, and stores its full contents all at once into a variable. Although this function cannot fully replace file:read(), it can be very handy from time to time!

Processing command-line arguments

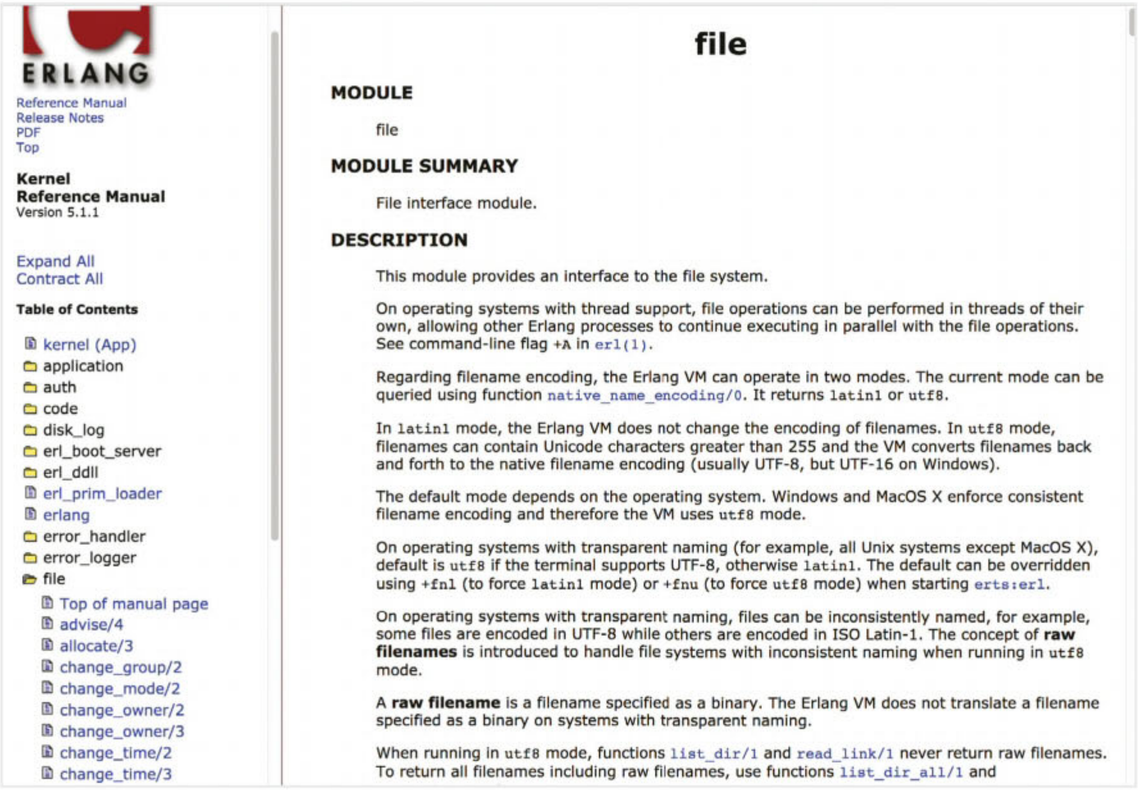
This section will illustrate how to read the command-line arguments of a program in Erlang, which is a very significant task because most of the time you get the names of the files you want to process as arguments. The Erlang code of **cmdArgs.erl** is below.

The following interaction with Erlang shell will help you clarify many things related to lists:

```
-module(cmdArgs).
-export([main/1]).

main(Args) ->
   io:format("Printing arguments:~n"),
   lists:foreach(fun(Arg) -> io:format("Arg: ~p~n",
[Arg]) end,Args).
```

Executing the main/0 function of **cmdArgs.erl** from the command line generates the following kind of output:

```
$ erlc cmdArgs.erl
$ erl -noshell -s cmdArgs main 1 two 3 4 five -s
init stop
Printing arguments:
Arg: '1'
Arg: two
Arg: '3'
Arg: '4'
Arg: five
```



ERLANG

Reference Manual
Release Notes
PDF
Top

Kernel
Reference Manual
Version 5.1.1

Expand All
Contract All

Table of Contents

- kernel (App)
- application
- auth
- code
- disk_log
- erl_boot_server
- erl_ddll
- erl_prim_loader
- erlang
- error_handler
- error_logger
- file
  - Top of manual page
  - advise/4
  - allocate/3
  - change_group/2
  - change_mode/2
  - change_owner/2
  - change_owner/3
  - change_time/2
  - change_time/3

file

MODULE

file

MODULE SUMMARY

File interface module.

DESCRIPTION

This module provides an interface to the file system.

On operating systems with thread support, file operations can be performed in threads of their own, allowing other Erlang processes to continue executing in parallel with the file operations. See command-line flag `+A` in `erl(1)`.

Regarding filename encoding, the Erlang VM can operate in two modes. The current mode can be queried using function `native_name_encoding/0`. It returns `latin1` or `utf8`.

In `latin1` mode, the Erlang VM does not change the encoding of filenames. In `utf8` mode, filenames can contain Unicode characters greater than 255 and the VM converts filenames back and forth to the native filename encoding (usually UTF-8, but UTF-16 on Windows).

The default mode depends on the operating system. Windows and MacOS X enforce consistent filename encoding and therefore the VM uses `utf8` mode.

On operating systems with transparent naming (for example, all Unix systems except MacOS X), default is `utf8` if the terminal supports UTF-8, otherwise `latin1`. The default can be overridden using `+fnl` (to force `latin1` mode) or `+fnu` (to force `utf8` mode) when starting `erts:erl`.

On operating systems with transparent naming, files can be inconsistently named, for example, some files are encoded in UTF-8 while others are encoded in ISO Latin-1. The concept of **raw filenames** is introduced to handle file systems with inconsistent naming when running in `utf8` mode.

A **raw filename** is a filename specified as a binary. The Erlang VM does not translate a filename specified as a binary on systems with transparent naming.

When running in `utf8` mode, functions `list_dir/1` and `read_link/1` never return raw filenames. To return all filenames including raw filenames, use functions `list_dir_all/1` and

Opening text files for reading

The following Erlang code opens a file that is identified by File for reading:

```
{ok, FD} = file:open(File,[read])
```

The use of **file:open()** is illustrated in **workText.erl**. Executing **lineByLine/1** from **workText.erl** generates the following kind of output:

```
1> c(workText).
{ok,workText}
2> workText:lineByLine("cmdArgs.erl").
-module(cmdArgs).
…
ok
```

The **lineByLine/1** function reads a file line by line and prints its contents with the help of the **read_line_by_line(FD)** function.

As you can see, opening a file for reading can fail for various reasons, including file not found and not having the right file permissions to open a file. So, if there is such an error somewhere, you will get the following kind of error message:

```
4> workText:lineByLine("cmdArgs").
** exception error: no match of right hand side
value {error,enoent}
   in function  workText:lineByLine/1 (workText.erl,
line 5)
```

Learning to interpret and work with error messages is critical and handy because it allows you to decide what to do depending on the kind of error you get. Please note that you are going to get the same error message when you are dealing with a binary file.

Figure 1 (overleaf) shows the implementation of both **lineByLine/1** and **read_line_by_line/1** functions as found in **workText.erl**. The **file:read_line()** function reads a single line from a file.

Opening text files for writing

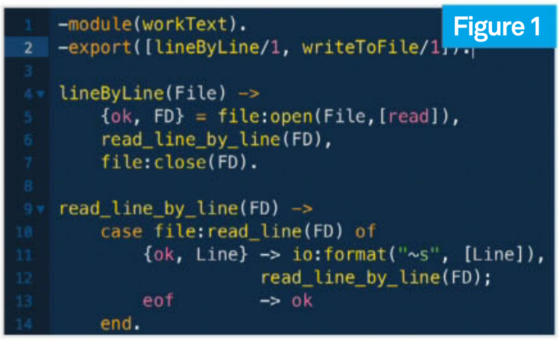
In order to open a file for writing, you will need to change the value of the second argument of the **file:open()** function, which is the Mode. If the file you are opening for writing already exists, it will be truncated, so be extra careful with it. If you do not want this behaviour, use the append mode.

Figure 2 shows the implementation of **writeToFile/1** that illustrates the use of the **file:write()** function. Executing **writeToFile/1** generates the following results:

```
1> c(workText).
{ok,workText}
2> workText:writeToFile("./testing").
ok
3> q().
ok
4> $ cat testing
A Message.
```

Error handling

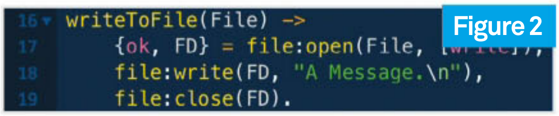
Error handling is a crucial task when you have to deal with file read and write operations because many things can go wrong during file I/O. Once again, error handling in



```
-module(workText).
-export([lineByLine/1, writeToFile/1]).

lineByLine(File) ->
    {ok, FD} = file:open(File,[read]),
    read_line_by_line(FD),
    file:close(FD).

read_line_by_line(FD) ->
    case file:read_line(FD) of
        {ok, Line} -> io:format("~s", [Line]),
                      read_line_by_line(FD);
        eof        -> ok
    end.
```

Figure 1

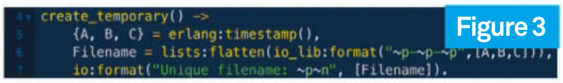
Right The use of the file:open/2 function for opening a file for reading and file:read_line/1 for reading a single line from a open file



```
writeToFile(File) ->
    {ok, FD} = file:open(File, [write]),
    file:write(FD, "A Message.\n"),
    file:close(FD).
```

Figure 2

Right The use of the file:write() Erlang function for writing data to a file



```
create_temporary() ->
    {A, B, C} = erlang:timestamp(),
    Filename = lists:flatten(io_lib:format("~p-~p-~p",[A,B,C])),
    io:format("Unique filename: ~p~n", [Filename]).
```

Figure 3

Right The Erlang code presented in this figure shows how you can create unique names

Erlang is implemented with the help of pattern matching. The following Erlang function presents a simple way to handle errors related to **file:open()**:

```
open_file_with_error_handling(File) ->
  case file:open(File, [read]) of
    {ok, Fd} ->
      io:format("File ~p opened successfully.~n",
[File]),
      file:close(Fd);
    {error, Reason} ->
      io:format("Error opening ~p.~n", [File]),
        io:format("Reason:~p.~n", [Reason])
  end.
```

You are also allowed to specifically handle some of the error messages. You can find the code inside **workBin.erl**. Executing it generates the following kind of output:

```
2> workBin:open_file_with_error_handling("wc.erl").
File "wc.erl" opened successfully.
ok
3> workBin:open_file_with_error_handling("wc").
Error opening "wc".
Reason:enoent.
Ok
```

Is Erlang good for text processing?

Erlang should not be your first choice for processing and parsing text files, even if you are a big fan of Erlang – programming languages such as AWK, Python, Perl and Ruby might be better candidates for such tasks because they need less code to do the job. The only rational reason to use Erlang for text manipulation is in case you have to process huge amounts of text, because Erlang has marvellous scaling capabilities.

Having said that, if you have a big Erlang application and you want to do a little text parsing without having to use another programming language, Erlang pattern matching will greatly assist you and make your life a lot easier.

You can learn about the various file-related error codes and their descriptions by visiting **bit.ly/2kMOHcM**.

Creating temporary files

This section will explain how to create a temporary file in Erlang using a random filename, which is pretty handy when you want to save your data without having to worry about accidentally selecting an existing filename. The developed function makes use of the **erlang:timestamp()** function that returns the current system time as a tuple with three elements, which is unique for each given system. The tuple is converted into a single variable that can be used for generating a new file or directory. **Figure 3** shows the Erlang code of the **create_temporary()** function that can be found in the **workBin.erl file**. Executing it generates the following results:

```
10> c(workBin).
{ok,workBin}
11> workBin:create_temporary().
Unique filename: "1485-326598-522782"
ok
12> workBin:create_temporary().
Unique filename: "1485-326600-978734"
ok
```

Developing wc in Erlang

This section will present a rudimentary implementation of **wc(1)** in Erlang. The code of **wc.erl** will not support command-line switches and will not print the total number of characters, words and lines read, because this would require too much code.

Figure 4 shows a part of the Erlang code of **wc.erl**. As you can understand from the implementation of **wc.erl**, Erlang might not be the best candidate for such command-line tools!

The input files are passed as a list to the **wc:start()** function in order to be processed. However, if you want to process a single file, then you can use the **wc:one_file()** function instead. The two most important functions of **wc.erl** are **process_file()** and **count_bin()**. The first one makes sure that all files are processed whereas the second one does the actual processing with a little help from **which_char()**. The contents of each file are processed character-by-character using list operations, which is pretty common in functional programming languages. Additionally, the definition of **which_char()** allows you to make easy changes to the way you treat special characters.

Executing **wc:start()** with a number of filenames as arguments creates the following kind of output:

```
2> wc:start(["wc.erl","mwc.erl"]).
file       chars          words          lines
wc.erl     1828           217            84
mwc.erl    1899           222            88
ok
```

If you modify the Erlang code of **wc.erl** a little, you are able to find out the flow of the program, which can be very practical

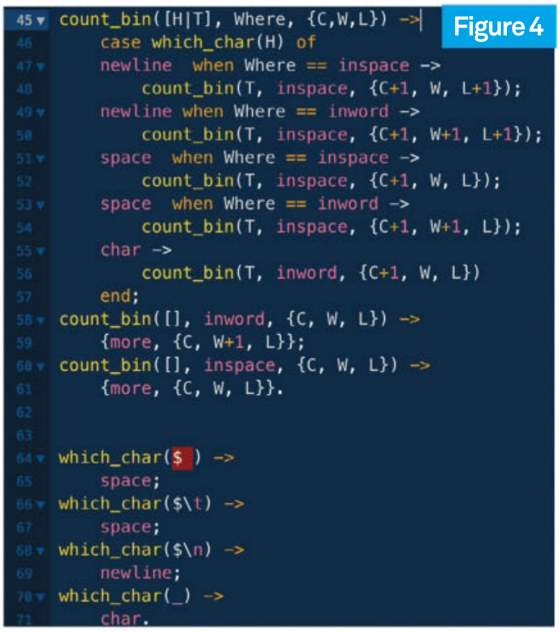


Figure 4

```
count_bin([H|T], Where, {C,W,L}) ->
    case which_char(H) of
    newline  when Where == inspace ->
        count_bin(T, inspace, {C+1, W, L+1});
    newline when Where == inword ->
        count_bin(T, inspace, {C+1, W+1, L+1});
    space  when Where == inspace ->
        count_bin(T, inspace, {C+1, W, L});
    space  when Where == inword ->
        count_bin(T, inspace, {C+1, W+1, L});
    char ->
        count_bin(T, inword, {C+1, W, L})
    end;
count_bin([], inword, {C, W, L}) ->
    {more, {C, W+1, L}};
count_bin([], inspace, {C, W, L}) ->
    {more, {C, W, L}}.


which_char($ ) ->
    space;
which_char($\t) ->
    space;
which_char($\n) ->
    newline;
which_char(_) ->
    char.
```

for debugging purposes. The simplest thing you can do is printing a given character each time you visit a function that interests you. Executing the modified version of **wc.erl**, which is named **mwc.erl**, generates the following kind of output:

```
10> mwc:start(["mwc.erl"]).
#_-+++++++++ …
```

Please note that the Erlang code of **wc.erl** is relatively advanced, so do not be disappointed if you do not understand all of it!

Working with binary files

Binary files are different to plain text files and therefore require a different approach to processing. The approach depends on the exact format of the binary file and therefore cannot be determined in advance.

You can open a binary file for reading in raw mode as follows:

```
1> {ok, S} = file:open("network.erl", [read,
binary, raw]).
{ok,{file_descriptor,prim_file,{#Port<0.435>,33}}}
```

The previous command gives you random access to the file you just opened. However, reading such a file requires the use of the **file:pread()** function.

Similarly, you can open a binary file for writing in raw mode as follows:

```
2> {ok, S} = file:open("aRAWbinary",
[raw,write,binary]).
{ok,{file_descriptor,prim_file,{#Port<0.435>,33}}}
```

Writing to a raw binary file can be done using the **file:pwrite()** function.

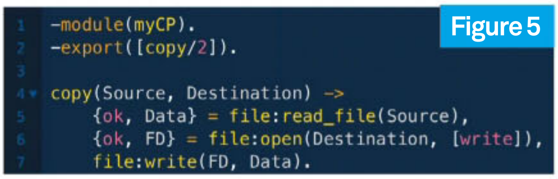


Figure 5

```
-module(myCP).
-export([copy/2]).

copy(Source, Destination) ->
    {ok, Data} = file:read_file(Source),
    {ok, FD} = file:open(Destination, [write]),
    file:write(FD, Data).
```

Talking more about raw binary files is beyond the scope of this tutorial.

Implementing cp in Erlang

The **cp(1)** utility reads and writes all files in binary format, and the Erlang implementation is no exception to this unofficial rule. The easiest way to implement the core functionality of **cp(1)** in Erlang is using the **file:read_file()** function. The relevant Erlang code can be found in **myCP.erl**, which you can also see in **Figure 5**. The module needs a small function to do the job because **file:read_file/1** reads the entire contents of a file and stores it into a single variable, whereas **file:write()** writes the contents of the variable to a file descriptor that was open for writing. Executing **myCP:copy/2** generates the following:

```
1> c(myCP).
{ok,myCP}
2> myCP:copy("myCP.erl", "copyMyCP.erl").
ok
$ diff copyMyCP.erl myCP.erl
```

The empty output of **diff** proves that **copyMyCP.erl** and **myCP.erl** are exactly the same!

Useful functions related to files

This section will illustrate various convenient Erlang methods that allow you to determine whether a file is really a file, whether a directory is in reality a directory, etc.

First, the **file:list_dir(directory)** function returns a list that contains the filenames of the given directory:

```
3> file:list_dir(".").
{ok,["myCP.erl","cmdArgs.erl","wc.erl","workBin.erl",
  "workText.erl"]}
```

The Mnesia database

Databases are related to files because they also provide a way of storing data. Mnesia is an extremely fast, distributed DBMS written in Erlang that is part of the standard Erlang distribution. You can query Mnesia with queries that are similar to SQL and List Comprehensions. As you might recall from a previous Erlang tutorial, List Comprehensions offer a mathematical way to create lists:

```
1> [ X || X <- [1,2,3,4,5,6] , X > 3].
[4,5,6]
```

The previous Erlang code creates a new list based on a mathematical condition.

Although a forthcoming tutorial will talk more about Erlang and Mnesia, you can learn more about Mnesia at: **erlang.org/doc/apps/mnesia/**.

Right The code of the workBin:visit() function that walks a directory tree in Erlang

Across How you can find information about a module from the Erlang shell. The output also includes the names of the functions that are exported

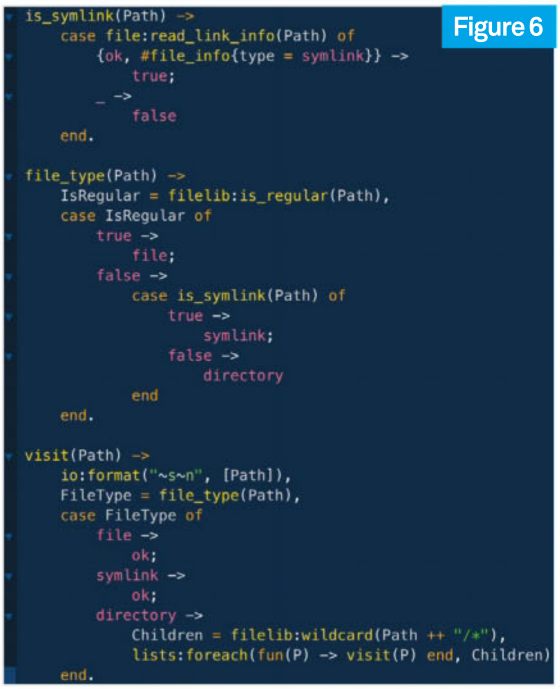


Figure 6

```
is_symlink(Path) ->
    case file:read_link_info(Path) of
        {ok, #file_info{type = symlink}} ->
            true;
        _ ->
            false
    end.

file_type(Path) ->
    IsRegular = filelib:is_regular(Path),
    case IsRegular of
        true ->
            file;
        false ->
            case is_symlink(Path) of
                true ->
                    symlink;
                false ->
                    directory
            end
    end.

visit(Path) ->
    io:format("~s~n", [Path]),
    FileType = file_type(Path),
    case FileType of
        file ->
            ok;
        symlink ->
            ok;
        directory ->
            Children = filelib:wildcard(Path ++ "/*"),
            lists:foreach(fun(P) -> visit(P) end, Children)
    end.
```

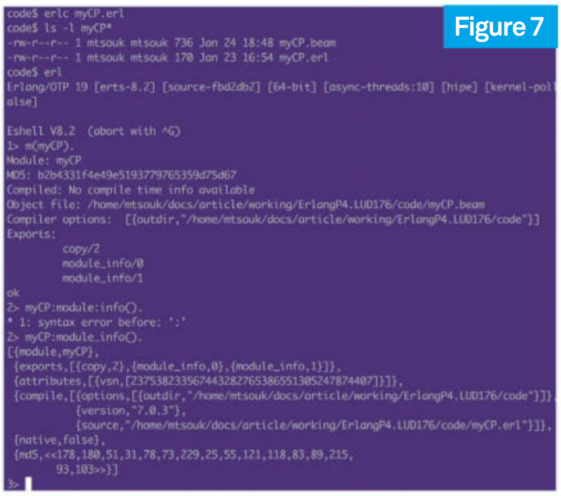


Figure 7

```
code$ erlc myCP.erl
code$ ls -l myCP*
-rw-r--r-- 1 mtsouk mtsouk 736 Jan 24 18:48 myCP.beam
-rw-r--r-- 1 mtsouk mtsouk 170 Jan 23 16:54 myCP.erl
code$ erl
Erlang/OTP 19 [erts-8.2] [source-fbd2db2] [64-bit] [async-threads:10] [hipe] [kernel-pol
alse]

Eshell V8.2  (abort with ^G)
1> m(myCP).
Module: myCP
MD5: b2b4331f4e49e5193779765359d75d67
Compiled: No compile time info available
Object file: /home/mtsouk/docs/article/working/ErlangP4.LUD176/code/myCP.beam
Compiler options:  [{outdir,"/home/mtsouk/docs/article/working/ErlangP4.LUD176/code"}]
Exports:
         copy/2
         module_info/0
         module_info/1
ok
2> myCP:module:info().
* 1: syntax error before: ':'
2> myCP:module_info().
[{module,myCP},
 {exports,[{copy,2},{module_info,0},{module_info,1}]},
 {attributes,[{vsn,[237538233567443282765386551305247874407]}]},
 {compile,[{options,[{outdir,"/home/mtsouk/docs/article/working/ErlangP4.LUD176/code"}]},
           {version,"7.0.3"},
           {source,"/home/mtsouk/docs/article/working/ErlangP4.LUD176/code/myCP.erl"}]},
 {native,false},
 {md5,<<178,180,51,31,78,73,229,25,55,121,118,83,89,215,
        93,103>>}]
3> 
```

The **filelib:file_size()** function returns the size of a file or a directory that is given as a parameter to it:

```
5> filelib:file_size(".").
4096
```

The **filelib:is_file()** and **filelib:is_dir()** functions enable you to determine whether the given path is a file or a directory, respectively:

```
6> filelib:is_file(".").
true
7> filelib:is_dir(".").
true
8> filelib:is_dir("wc.erl").
false
9> filelib:is_file("wc.erl").
true
10> filelib:is_dir("/dev/random").
false
11> filelib:is_file("/dev/random").
False
```

As you can see, **filelib:is_file()** considers directories as files but can differentiate between regular files and special files such as **/dev/random**.

Visiting all files in a directory tree

There is a very frequent need to visit all files in a directory and its subdirectories. This section will teach you how to perform this task in Erlang. Once again, the developed function requires one argument, which is a directory, and returns the desired information. The **visit()** function is included in the **workBin.erl** file. Executing **visit()** generates the following kind of output:

```
3> workBin:visit("/var/spool").
/var/spool
/var/spool/cups
/var/spool/mqueue
…
```

You can see the related Erlang code in **Figure 6**. The most important Erlang function in the implementation of **visit()** is **filelib:wildcard()**, which returns the list of files that match the given criteria.

The **file.hlr** file holds the definition of the **file_info** record – the information in **file_info** allows you to determine the type of file you are dealing with. On a Debian Linux system, the **file.hrl** file can be found at **/usr/lib/erlang/lib/kernel-5.1.1/include**. You can find more information about the data that **file_info** holds at: **erlang.org/doc/man/file.html**.

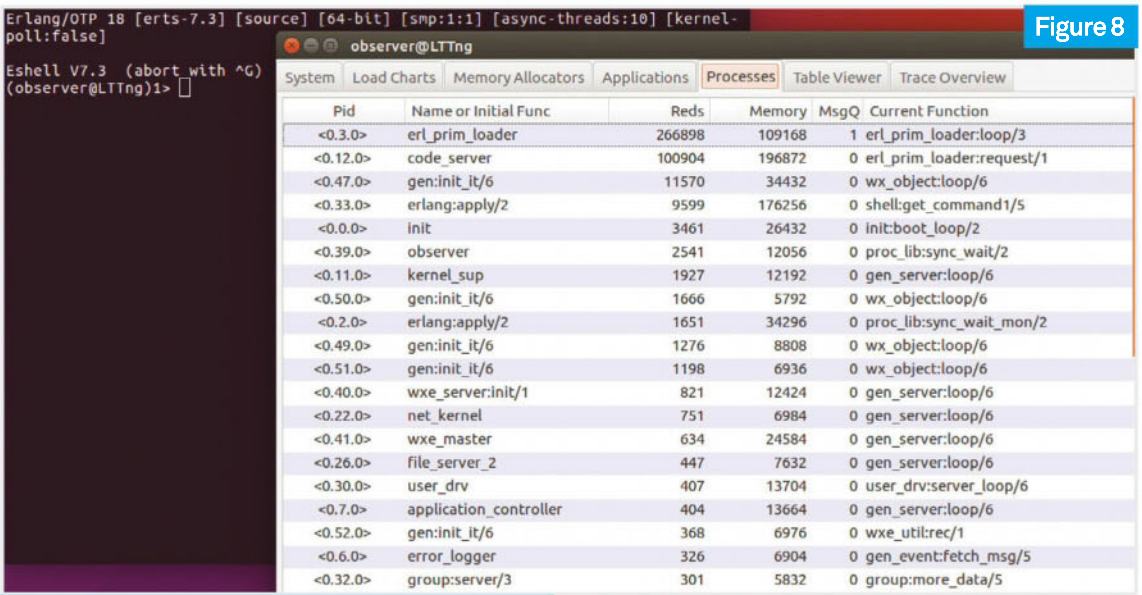
Finding information about files

The following code illustrates how you can find more information about files using the **file_info** record:

```
file_permissions(Path) ->
    case file:read_file_info(Path) of
        {ok, FR} -> io_lib:format("~.8B",
[FR#file_info.mode band 8#777]);
    _ ->
        error
    end.
```

The previous code prints the file permissions of a file or a directory. Please bear in mind that although **file:read_file_info()** contains information about file size and file type, calling **file_size()** and **is_dir()** is more practical than using **file:read_file_info()**. However, note that some of the data returned by **file:read_file_info()** cannot be discovered using other Erlang functions.

Executing **file_permissions()** generates the next kind of output:



```
Erlang/OTP 18 [erts-7.3] [source] [64-bit] [smp:1:1] [async-threads:10] [kernel-
poll:false]

Eshell V7.3  (abort with ^G)
(observer@LTTng)1> 
```

Pid	Name or Initial Func	Reds	Memory	MsgQ	Current Function
<0.3.0>	erl_prim_loader	266898	109168	1	erl_prim_loader:loop/3
<0.12.0>	code_server	100904	196872	0	erl_prim_loader:request/1
<0.47.0>	gen:init_it/6	11570	34432	0	wx_object:loop/6
<0.33.0>	erlang:apply/2	9599	176256	0	shell:get_command1/5
<0.0.0>	init	3461	26432	0	init:boot_loop/2
<0.39.0>	observer	2541	12056	0	proc_lib:sync_wait/2
<0.11.0>	kernel_sup	1927	12192	0	gen_server:loop/6
<0.50.0>	gen:init_it/6	1666	5792	0	wx_object:loop/6
<0.2.0>	erlang:apply/2	1651	34296	0	proc_lib:sync_wait_mon/2
<0.49.0>	gen:init_it/6	1276	8808	0	wx_object:loop/6
<0.51.0>	gen:init_it/6	1198	6936	0	wx_object:loop/6
<0.40.0>	wxe_server:init/1	821	12424	0	gen_server:loop/6
<0.22.0>	net_kernel	751	6984	0	gen_server:loop/6
<0.41.0>	wxe_master	634	24584	0	gen_server:loop/6
<0.26.0>	file_server_2	447	7632	0	gen_server:loop/6
<0.30.0>	user_drv	407	13704	0	user_drv:server_loop/6
<0.7.0>	application_controller	404	13664	0	gen_server:loop/6
<0.52.0>	gen:init_it/6	368	6976	0	wxe_util:rec/1
<0.6.0>	error_logger	326	6904	0	gen_event:fetch_msg/5
<0.32.0>	group:server/3	301	5832	0	group:more_data/5

Figure 8

Left One of the tabs of the Erlang Observer application in action

```
2> workBin:file_permissions("wc.erl").
["644"]
3> workBin:file_permissions("/tmp").
["777"]
```

Finding information about modules

There are two easy ways to find information about the exported functions of an existing module. So, having a module named **aModule**, you can use either of the following two commands to learn more about it:

```
1> m(aModule).
2> aModule:module_info().
```

Please note that you need the BEAM file of the module in order to execute any one of these two commands. **Figure 7** shows the output of the aforementioned commands.

Watching Erlang processes

Erlang provides a powerful graphical application named Observer that allows you to inspect the running Erlang processes. You can start the Observer application by executing the following command from the Erlang shell:

```
1> observer:start().
ok
```

However, you can also start Observer from the Linux command line – the advantage of this approach is that it uses fewer processes and requires less CPU power:

```
$ erl -sname observer -hidden -setcookie
MyCookie -run observer
```

Figure 8 shows the Erlang Observer application, which has replaced the Pman application. You can find more information about Erlang Observer by visiting **erlang.org/doc/apps/observer/observer_ug.html**. A forthcoming tutorial will talk more about the Observer.

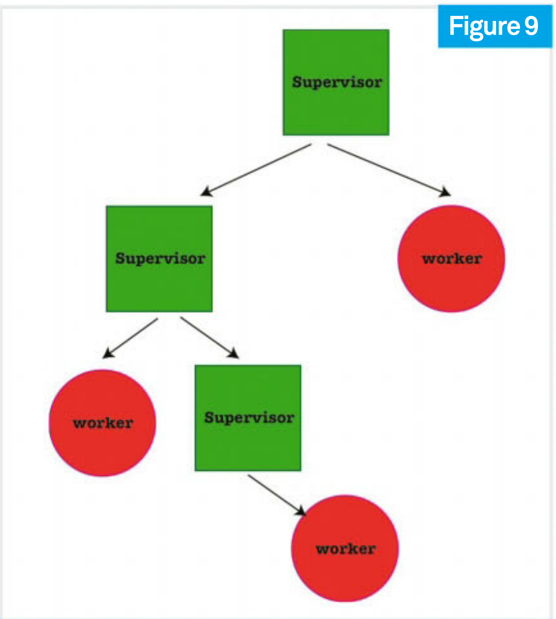


Figure 9

Left A plot of a simple supervision tree that helps you understand the relationships between workers and supervisors

More about OTP

A supervision tree is a graphical way of representing the relationships between the workers and the supervisors of an application. Workers are the Erlang processes that do the real work whereas supervisors are the Erlang processes that keep an eye on the status of workers or other supervisors. There are three main types of children in a supervision tree: permanent, which get started when they die; transient, which only get started if they die abnormally; and temporary children that never get started when they die.

Figure 9 shows the plot of a supervision tree. The main advantage you get by constructing and looking at a supervision tree is a quick understanding of the architecture of your application.

You will see more supervision tree examples when we talk about OTP.

Server administration in Ubuntu

Take a deeper look at server admin under Ubuntu

Swayam Prakasha
has a master's degree in computer engineering. He has been working in information technology for several years, concentrating on areas such as operating systems, networking, network security, electronic commerce, internet services, LDAP and web servers. Swayam has authored a number of articles for trade publications, and he presents his own papers at industry conferences. He can be reached at swayam.prakasha@gmail.com

Resources

System admin topics
bit.ly/2jn1syX

A step-by-step approach to understanding the Ubuntu Linux system
bit.ly/2kjCe42

Basic Ubuntu server administration
bit.ly/2kGsler

Ubuntu server guide (PDF)
bit.ly/18NkP9K

When we set up a Linux system to act as a server, we need to focus on several tasks. As the name implies, servers exist to serve. The data that they serve can include webpages, files, database information and so on. From a server administration perspective, there are some typical challenges to your system administration skills.

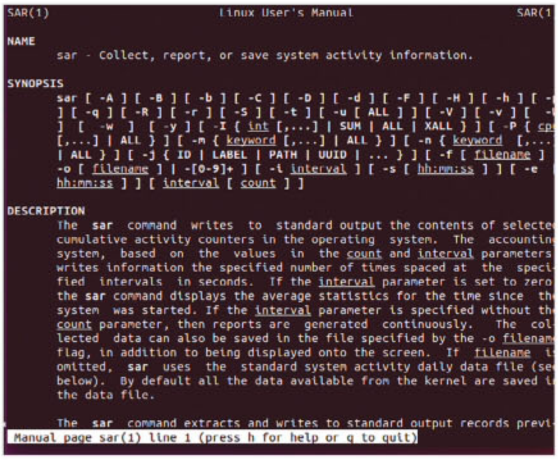
The first is remote access. In the majority of cases, administration will be carried out using remote access tools. Since graphical interfaces may not be available, you need to depend solely on command-line tools to carry out various tasks such as remote login, remote copying and remote execution. The most common of these tools are built on the SSH facility.

The second is security. As expected, a server will accept requests from remote users and systems. As a result, a server administrator will open the ports to the services that are needed and lock down the ports that are not required. Various tools are available that can be used to secure the services.

The third is constant monitoring. Servers are expected to stay up 24-7, 365 days a year. There is a significant need to configure the tools that can be used to monitor each of your servers. With such monitoring, you will be able to collect the required log messages and also trigger an email whenever suspicious activity occurs. It is always good to collect data around CPU usage, memory usage, network activities and so on.

It is important to note here that whether you are starting a file server or a web server, many of the steps that are required to get the server up and running will remain the same. When it comes to configuration and tuning, we see a bunch of differences. Once we have successfully installed the server, the next step is to have a look at its default configuration. It needs to be noted here that most of the server packages are installed with a default configuration setting that leans towards providing more security.

As we know, most Linux servers can be configured using plain text in the **/etc** directory. In most cases, there is a primary configuration file and sometimes, a related configuration directory from which files with an extension of .conf can be pulled into the main configuration file. Apache web server is an example where we have a primary configuration file and a directory where other configuration files can be dropped in and included with the service. The main configuration file is **/etc/httpd/conf/httpd.conf** and the configuration directory is **/etc/httpd/conf.d/**.



```
SAR(1)                    Linux User's Manual                    SAR(1

NAME
       sar - Collect, report, or save system activity information.

SYNOPSIS
       sar [ -A ] [ -B ] [ -b ] [ -C ] [ -D ] [ -d ] [ -F ] [ -H ] [ -h ] [ -
       ] [ -q ] [ -R ] [ -r ] [ -S ] [ -t ] [ -u [ ALL ] ] [ -V ] [ -v ] [  -
       ]  [  -w  ]  [ -y ] [ -I { int [,...] | SUM | ALL | XALL } ] [ -P { cp
       [,...] | ALL } ] [ -m { keyword [,...] | ALL } ] [ -n { keyword  [,...
       | ALL } ] [ -j { ID | LABEL | PATH | UUID | ... } ] [ -f [ filename ]
       -o [ filename ] | -[0-9]+ ] [ -i interval ] [ -s [ hh:mm:ss ] ] [ -e
       hh:mm:ss ] ] [ interval [ count ] ]

DESCRIPTION
       The  sar  command  writes  to  standard output the contents of selecte
       cumulative activity counters in the operating  system.  The  accountin
       system,  based  on  the  values  in  the count and interval parameters
       writes information the specified number of times spaced at  the  speci-
       fied  intervals  in seconds.  If the interval parameter is set to zero
       the sar command displays the average statistics for the time since  th
       system  was started. If the interval parameter is specified without th
       count parameter, then reports are  generated  continuously.  The  col-
       lected  data can also be saved in the file specified by the -o filenam
       flag, in addition to being displayed onto the screen.  If  filename  i
       omitted,  sar  uses  the  standard system activity daily data file (se
       below).  By default all the data available from the kernel are saved i
       the data file.

       The  sar  command extracts and writes to standard output records previ-
 Manual page sar(1) line 1 (press h for help or q to quit)
```

Right The widely used sar utility for server administration

As mentioned earlier, most server software packages are installed with very minimal configuration and this has its own limitations if we need to fully use those servers. Mail servers and DNS servers are typical examples of servers with limited functionality. Please note that both of these servers are installed with their default configurations and they start on reboot. But the thing here is both of them only listen for the requests on your localhost. Therefore, until we configure these servers, people who are not logged into your local server will not be able to send email to that server or use your computer as a public DNS server.

Once we have installed and configured the server, the next step is to start it. Generally, most servers that we install on Linux are configured to start up when the system boots up. After that, the services run continuously and keep listening for requests. This will continue until the system is shut down. You can manage the services using two major facilities: system and System V init scripts. Regardless of which facility is used on the Linux system, it is up to you to determine whether you want the services to come up when the system boots and to start, stop and reload the services as needed. Another thing to note here is that the majority of the services are implemented as daemon processes. The table here gives a brief description of the parameters.

Parameter	Description
User and group permissions	Daemon processes normally run as users and groups other than the root. The reason behind this is that is if someone hacks these daemons, they will not have permission to access files beyond those the service can access.
Daemon configuration files	Generally, a service has a configuration file for the daemon, stored in **/etc/sysconfig**. Refer to the man page details of the specific daemon for more.
Port numbers	Most standard services have specific port numbers that daemons listen to and clients can connect to. Typically we do not need to change the ports that a daemon process listens on.

There is no mandate that all services need to run as daemon services. Some services run on demand by using the xinetd super server. We can consider on-demand services as the primary way of running always-available services. It is important to note here that the on-demand services will not run continuously, thereby listening for the requests. These services are registered with the xinetd daemon and when the requests come to the xinetd daemon for a service, the xinetd daemon launches the requested service and hands over the request to that service. There is a clear-cut advantage with this approach as we will have fewer daemon processes running.

You need to be very cautious when it comes to securing a server. On Linux systems, various measures can help to protect your servers and services from external attacks.

Having good passwords and password policies is the first line of defence when it comes to securing the Linux system. The best practice to follow here is to disallow a direct login by root and force everyone to log in as a regular user and then use sudo to become root. As another best practice, you can use the pluggable authentication module (PAM) facility, which will help adjust the number of times someone can have failed login attempts before access is blocked to that user.

The iptables firewall service is another security layer. Using this service, you can drop or reject every packet making requests for services on your system, except for those few that you have enabled. You can also inform iptables to allow service requests only from certain IP addresses and deny requests from other addresses.

You can allow or deny access to services that have TCP wrappers enabled by using the files **/etc/hosts.allow** and **/etc/hosts.deny** functions. Access can be allowed or denied based on the host name or IP address.

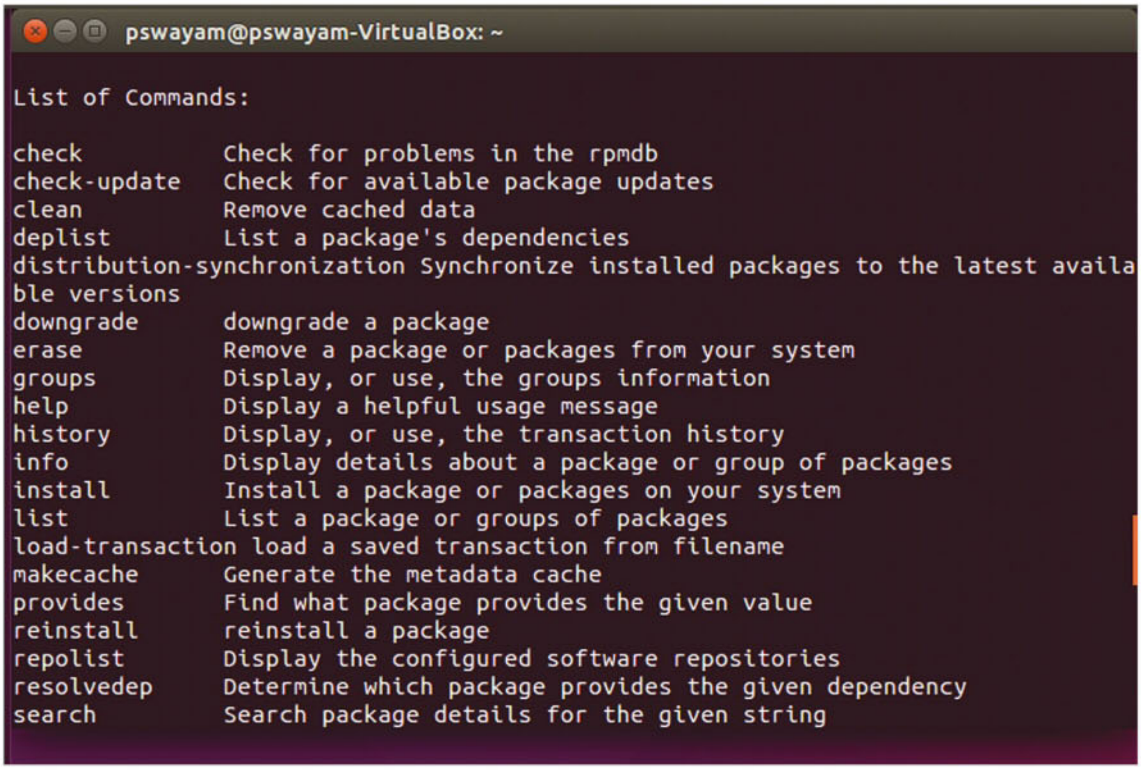
Most distributions nowadays come with the Security Enhanced Linux (SELinux) feature and by default, it is included in Enforcing mode. An important function of SELinux is that it will help protect the contents of your Linux system from the processes running on that system.

As another security measure, you can also work on security settings in configuration files. Within the

System logging

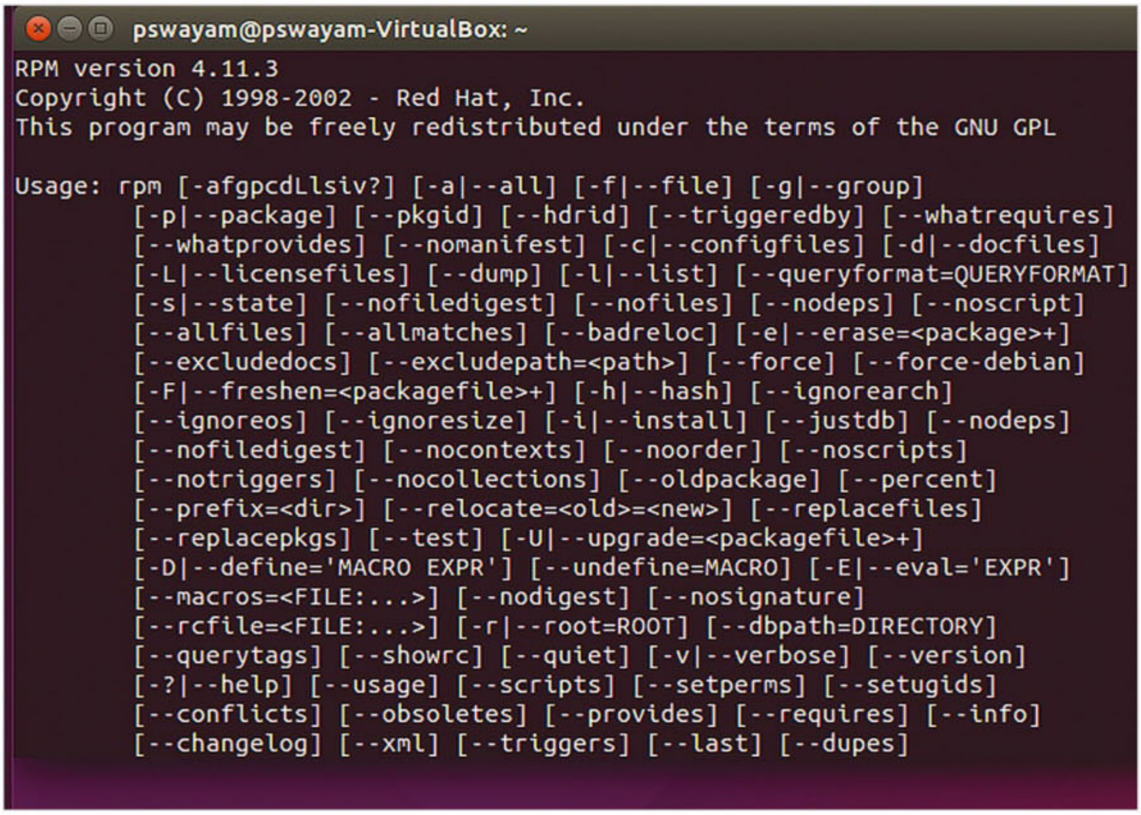
System logging is one of the basic services configured on a Linux system and it will help in keeping track of what is happening on the system. The rsyslog service will provide the features to gather the log messages from the software/applications that are running on the Linux box. A server administrator can then redirect these messages to local log files, devices or remote logging hosts.

> “Servers exist to serve. The data that they serve can include webpages, files and database information”


pswayam@pswayam-VirtualBox: ~

```
List of Commands:

check          Check for problems in the rpmdb
check-update   Check for available package updates
clean          Remove cached data
deplist        List a package's dependencies
distribution-synchronization Synchronize installed packages to the latest availa
ble versions
downgrade      downgrade a package
erase          Remove a package or packages from your system
groups         Display, or use, the groups information
help           Display a helpful usage message
history        Display, or use, the transaction history
info           Display details about a package or group of packages
install        Install a package or packages on your system
list           List a package or groups of packages
load-transaction load a saved transaction from filename
makecache      Generate the metadata cache
provides       Find what package provides the given value
reinstall      reinstall a package
repolist       Display the configured software repositories
resolvedep     Determine which package provides the given dependency
search         Search package details for the given string
```

Left Various options with the yum command



```
RPM version 4.11.3
Copyright (C) 1998-2002 - Red Hat, Inc.
This program may be freely redistributed under the terms of the GNU GPL

Usage: rpm [-afgpcdLlsiv?] [-a|--all] [-f|--file] [-g|--group]
        [-p|--package] [--pkgid] [--hdrid] [--triggeredby] [--whatrequires]
        [--whatprovides] [--nomanifest] [-c|--configfiles] [-d|--docfiles]
        [-L|--licensefiles] [--dump] [-l|--list] [--queryformat=QUERYFORMAT]
        [-s|--state] [--nofiledigest] [--nofiles] [--nodeps] [--noscript]
        [--allfiles] [--allmatches] [--badreloc] [-e|--erase=<package>+]
        [--excludedocs] [--excludepath=<path>] [--force] [--force-debian]
        [-F|--freshen=<packagefile>+] [-h|--hash] [--ignorearch]
        [--ignoreos] [--ignoresize] [-i|--install] [--justdb] [--nodeps]
        [--nofiledigest] [--nocontexts] [--noorder] [--noscripts]
        [--notriggers] [--nocollections] [--oldpackage] [--percent]
        [--prefix=<dir>] [--relocate=<old>=<new>] [--replacefiles]
        [--replacepkgs] [--test] [-U|--upgrade=<packagefile>+]
        [-D|--define='MACRO EXPR'] [--undefine=MACRO] [-E|--eval='EXPR']
        [--macros=<FILE:...>] [--nodigest] [--nosignature]
        [--rcfile=<FILE:...>] [-r|--root=ROOT] [--dbpath=DIRECTORY]
        [--querytags] [--showrc] [--quiet] [-v|--verbose] [--version]
        [-?|--help] [--usage] [--scripts] [--setperms] [--setugids]
        [--conflicts] [--obsoletes] [--provides] [--requires] [--info]
        [--changelog] [--xml] [--triggers] [--last] [--dupes]
```

Right Usage of the rpm command

configuration files, you will be able to set the values so that services can be further secured.

Monitoring servers is a critical aspect. Since you will not be able to monitor every service every minute of every day, it is important to put in place some monitoring tools so that servers can be watched. This makes the life of a system administrator easier as they can find out when something needs their urgent focus. There are several tools that can be used for monitoring servers.

Using the rsyslog service, you will be able to collect important information and error conditions about various services. By default, all the log messages are directed into the log files located in the /var/log directory. From an improved security and convenience perspective, you can also redirect all log messages to a centralised server.

Sar is an important utility that can be configured to watch the activities that are happening on your system. Memory usage, CPU usage and network activities are some of those that can be monitored by sar.

It is very important to keep system software up to date all the time. In other words, we need to ensure that the updated software packages containing the patches are installed on your system. One command that's widely used is yum – it will check for and install the packages that are available for your system

It is also important to check your file systems so that you can identify the possible intrusions. This can be done by using the following command:

```
~$ rpm -V
```

This command will help us check if any commands, documents or configuration files have been tampered with on the system.

The server administrator depends heavily on Secure Shell (SSH) services for managing remote access to the servers. The Secure Shell tools can be considered as a set of client and server applications and they help in carrying out the basic communications between the client machines and the Linux server. The popular SSH tools are ssh, scp and sftp. It can be noted here that as the communication between the clients and the server is encrypted, these tools are more secure. With these tools, in addition to the communication, the authentication process is also encrypted. We know that communications from telnet and the earlier 'r' commands expose the data and passwords so that someone can easily access those over the network.

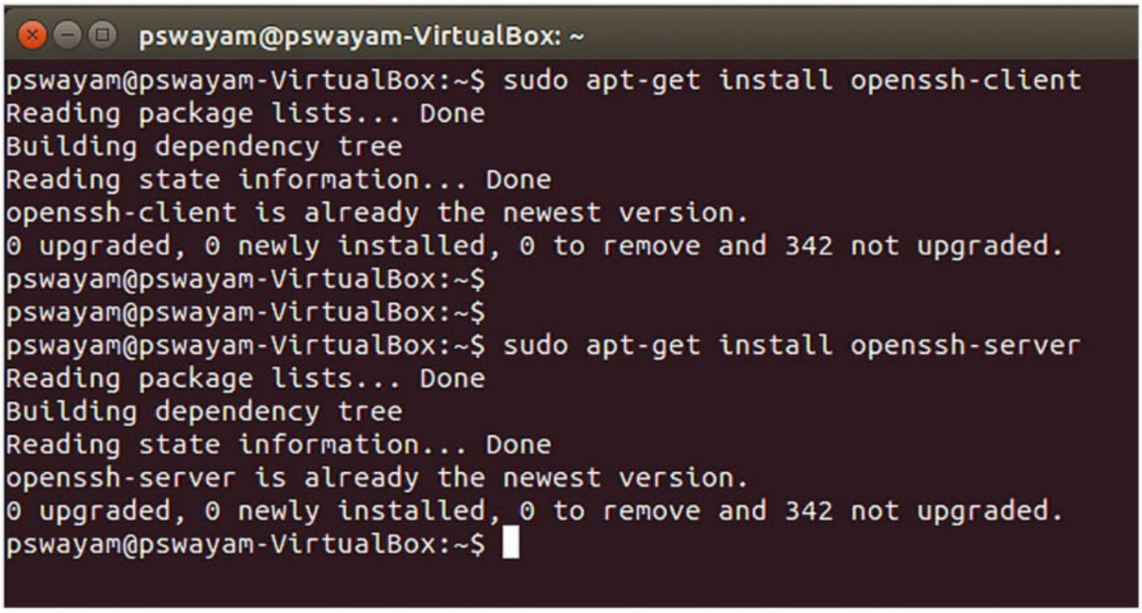
It can be noted here that most Linux systems include shell clients and may also include a Secure Shell server. The client and server software packages that contain these tools are openssh, openssh-clients and openssh-server.

You can use the following command for this purpose:

```
~$ yum list installed | grep ssh
```

If the server package is not installed, use the following command to install it:

```
~$ sudo apt-get install openssh-server
```



```
pswayam@pswayam-VirtualBox: ~
pswayam@pswayam-VirtualBox:~$ sudo apt-get install openssh-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-client is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 342 not upgraded.
pswayam@pswayam-VirtualBox:~$
pswayam@pswayam-VirtualBox:~$
pswayam@pswayam-VirtualBox:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 342 not upgraded.
pswayam@pswayam-VirtualBox:~$
```

Above Installing openssh-client and openssh-server

Linux systems that come with the openssh-server package already installed sometimes may not be configured so that they can start automatically. In such cases, you may need to use commands so that SSH server daemon (sshd) is up and running on your Linux system. But on Ubuntu, when we install openssh-server, the sshd daemon is always configured to start automatically at bootup. We may need to handle further configurations for the sshd daemon – this can be done by editing the /etc/ssh/sshd_config file. One configuration that is of critical importance here is to change PermitRootLogin setting from yes to no. This way, we will stop anyone remotely logging in as root.

Of all the Secure Shell tools, the most commonly used one is the ssh command. The server administrator can use this command for remote login, remote execution and other similar tasks. Commands such as scp and rsync can be used to copy one or more files between the SSH client and server system. By default, the Secure Shell tools authenticate using the standard Linux usernames and passwords, all done over an encrypted connection. These tools also support key-based authentication and this can be used to configure passwordless authentication between the clients and the SSH server. By using the ssh command, a server administrator can check that they will be able to log in to the Linux system running on the sshd service. Once you are logged in to the remote system, you can begin executing the shell commands. Please note that the connection functions like a normal login – the only difference here is that the data is encrypted as it travels over the network. Once we are done, we can type `exit` and that will end our remote connection.

The ssh command can also be used for remote execution. We can use this command to execute a command on the remote system and return the output to the local system. When we run a remote execution command with ssh that includes options or arguments, we need to ensure that the whole remote command line is surrounded by quotes.

Server administrators use scp and rsync commands for copying files between systems. The scp command is very similar to the older rcp command, the only difference being with scp that all communications are encrypted. You can copy the files from remote to local systems or vice versa. The scp command can also be used to back up the files and directories over a network. But when we compare scp with the rsync command, we can note that rsync is considered to be a better backup tool.

> "The server administrator depends heavily on Secure Shell services for managing remote access to the servers"

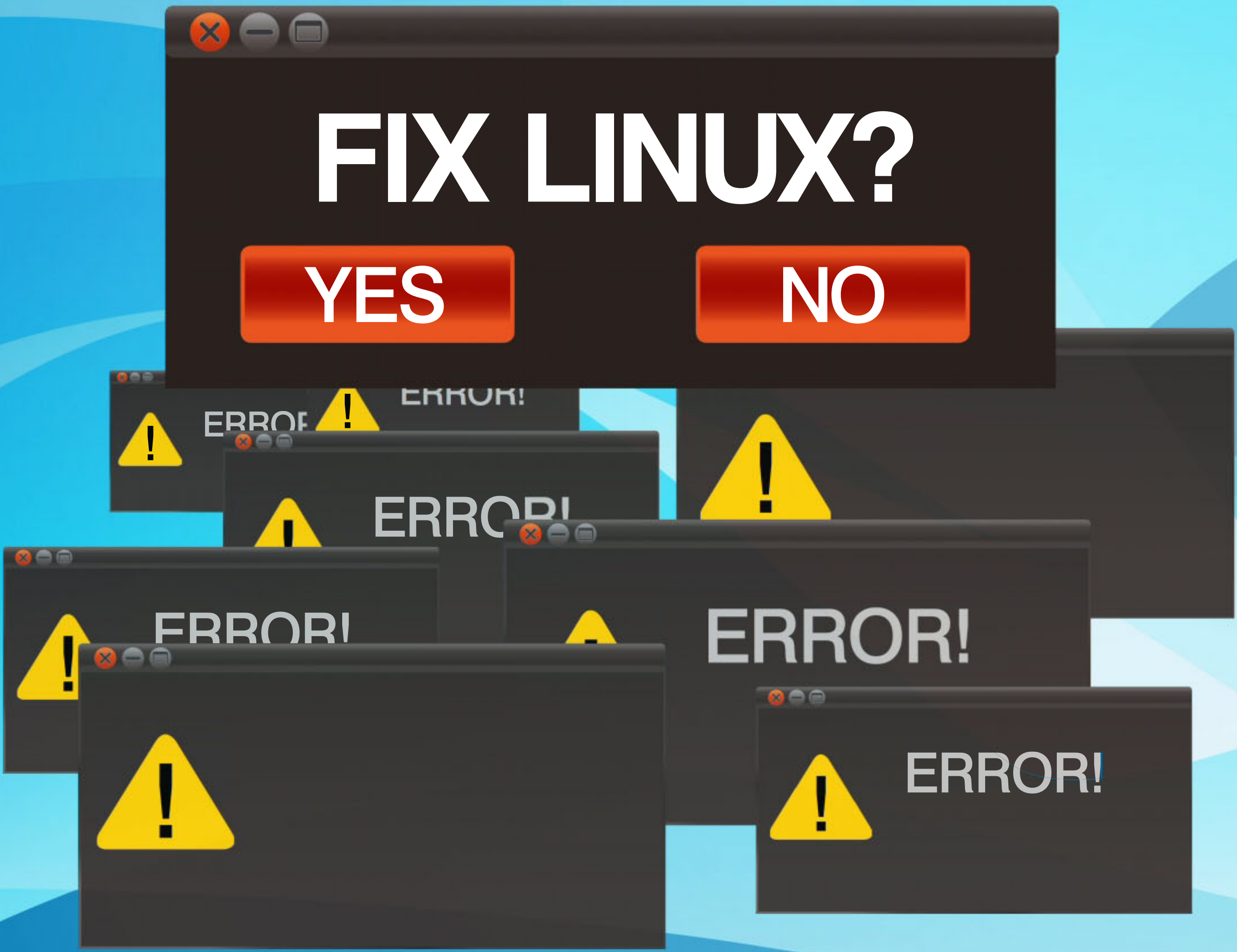
Watch the logs with logwatch

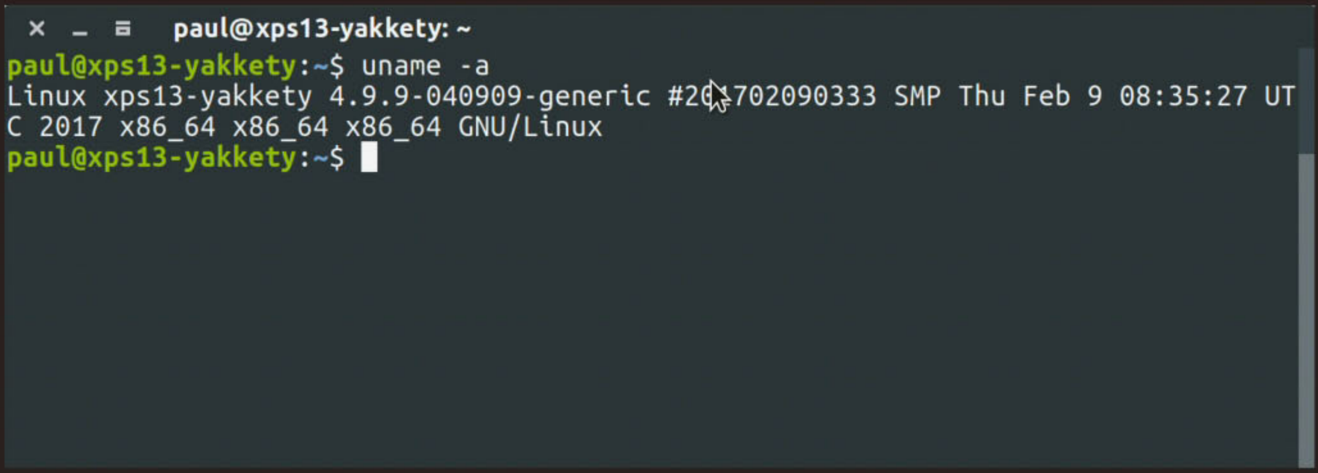
The logwatch service runs on most Linux systems. Once every night, this service will gather the messages that look like they might represent a problem. Then it will put them into an email message and send it to an administrator. Typically, the logwatch service runs from a cron job. In order to install this facility, you can use the following command:

```
~$ sudo yum install logwatch
```

35 ESSENTIAL FIXES

Whether you are a seasoned Linux user or a newcomer, you are likely to come across problems now and again. Here's how to solve them





```
paul@xps13-yakkety: ~
paul@xps13-yakkety:~$ uname -a
Linux xps13-yakkety 4.9.9-040909-generic #201702090333 SMP Thu Feb 9 08:35:27 UT
C 2017 x86_64 x86_64 x86_64 GNU/Linux
paul@xps13-yakkety:~$
```

Linux is a weird and wonderful beast. One of the best things about it is that it really gives you the chance to dig deep into the operating system to get everything up and running exactly as you like it. The flip side of this, however, is that as you tweak things, you are likely to come across unexpected problems, whether they're related to the distribution you are using, the hardware you're running, a particularly obscure piece of open source software or perhaps that Linux just works a little differently to what you are used to. This is particularly likely if you are using Linux for the first time after coming from Windows – it can be daunting at first in a lot of areas. This article highlights some issues you might come across when using Linux, whether you are just starting out or are a long-time Linux user.

> “As you tweak things in Linux, you are likely to come across unexpected problems”

If you come across a problem that's not featured in our guide then fear not, you're almost certainly not the first. Search engines can normally point you in the right direction for general queries, or if your issue is specific to a certain piece of software, the support forums or issue tracker for that product are a great place to start. The Linux community is a great bunch of people!

KERNEL ISSUES

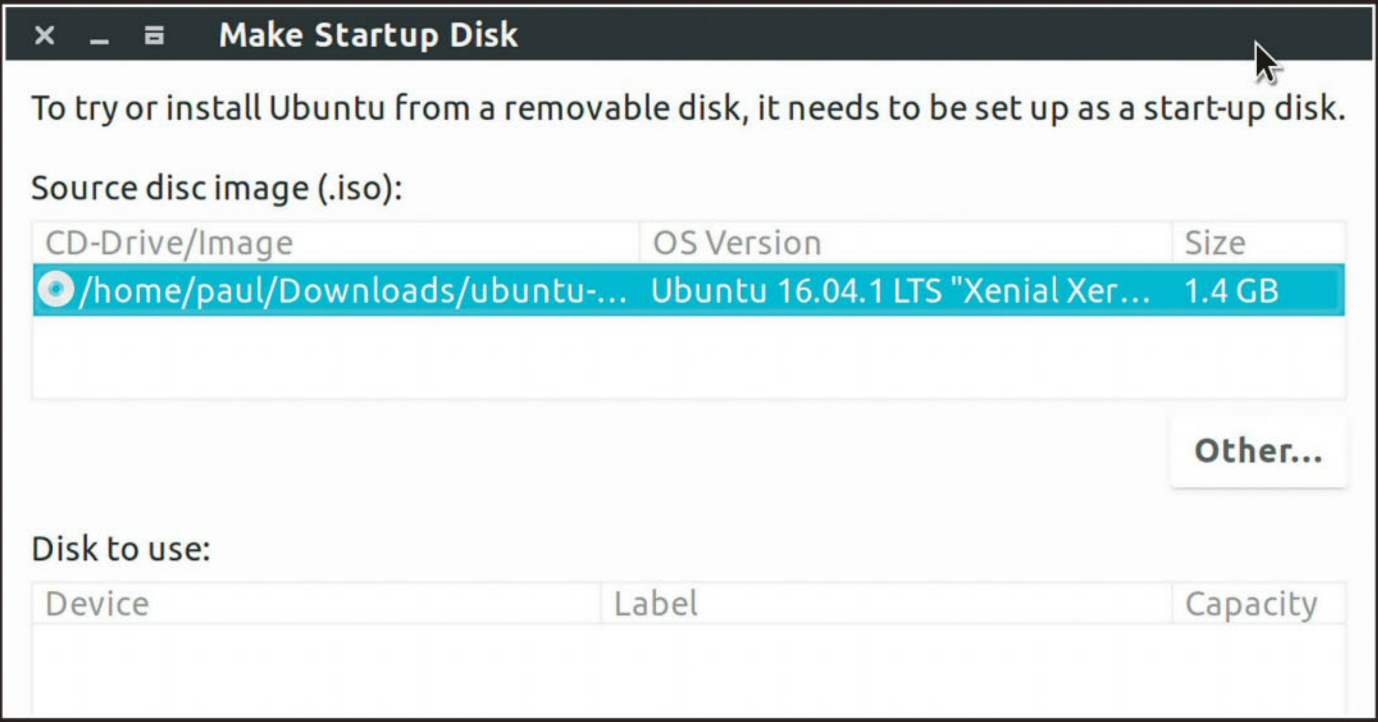
The Linux kernel is one of the largest open source projects in the world, with over 13 million lines of code – and it underpins everything your Linux machine does. Generally speaking, most Linux users don't think about the kernel too much; it just sits there, doing its thing. The kernel becomes most important and interesting to users when a forthcoming release adds support for a new piece of hardware that is in a machine or fixes some issues in existing hardware.

Typically, a specific kernel version is tied to a specific distribution release. For example, Ubuntu Xenial Xerus includes kernel 4.4, while Ubuntu Yakkety Yak includes kernel 4.8. Although this is the case, there is nothing to stop you installing a new version of the kernel on your system if you want to take advantage of some new features. One of the nicest things about the kernel is you can easily install a new version, switch between the two (via GRUB) and roll back to a previous release if needed.

The easiest way to install a newer kernel is using a package from the Ubuntu upstream kernels archive (**bit.ly/1zcskza**). Browse to the kernel version you want to install, select the appropriate architecture and download the file. Install as normal and reboot to load the new kernel. If you do need to revert, open the GRUB advanced menu, load the old kernel and uninstall the package in the normal way. Note that many users treat a kernel update like a BIOS update – if it ain't broke, don't fix it!

INSTALLATION AND UPGRADE ISSUES

Don't fall at the first hurdle on your Linux adventure – overcome these common install or upgrade problems



Make Startup Disk

To try or install Ubuntu from a removable disk, it needs to be set up as a start-up disk.

Source disc image (.iso):

CD-Drive/Image	OS Version	Size
/home/paul/Downloads/ubuntu-...	Ubuntu 16.04.1 LTS "Xenial Xer...	1.4 GB

Disk to use:

Device	Label	Capacity

Above You need to make the ISO disk image bootable

> “One of the most common issues is disks being ‘invisible’ to Linux installers. Fortunately, there is an easy fix”

1 THE INSTALL MEDIA WON'T BOOT

So you've downloaded Linux and either burned it to a CD/DVD or copied it to a USB stick. But it won't boot. Not a good start. First up, ensure that if you're using a CD/DVD, you've burned the actual ISO and not just copied the file. In the same way, if you're using USB, you need to follow a guide on how to make the image bootable, not just copy it to a stick. If you've done this, the next thing you need to do is access the boot sequence menu on your machine. This is pretty well hidden away on modern machines and the sequence varies between manufacturers. Try the **Esc** key or **F12** to display the selections.

2 THE INSTALLER DOESN'T SEE MY HARD DISK

Over recent years there have been a lot of changes to how PCs work, as we've moved from the BIOS era to the UEFI era. In addition, hard disks (and particularly SSDs) and how they communicate inside machines has progressed quickly. Many of these optimisations have been designed specifically for Windows, which can cause problems for Linux. One of the most common issues is disks being 'invisible' to Linux installers. Fortunately, there is an easy fix should you encounter this problem – in your computer's settings, change the hard disk mode to AHCI/SATA. This is simple enough, but making this change will likely stop your Windows install from starting up if you are dual-booting. To fix this, start Windows as normal and use the command:

```
bcdedit /set {current} safeboot minimal
```

...before changing the setting. Then, after making the change and completing the first boot, use:

```
bcdedit /deletevalue {current} safeboot
```

...to get back up and running.

3 THE PC STILL BOOTS INTO WINDOWS

A common problem when you install Linux on a new machine is that you'll complete the wizard, reboot as instructed, but then the computer still boots into Windows. What gives?! Don't worry, you haven't wasted your time – this is generally down to GRUB, the Linux bootloader, not installing correctly, typically not on the right disk. To resolve this issue, simply install and run Boot-Repair if it is available in your OS repo. If not, use:

```
sudo grub-install
```

...together with the appropriate disk, as displayed by using:

```
ls -l /dev/disk/by-label/
```

4 I DON'T HAVE WORKING WI-FI

Although Linux is supported on a vast range of hardware nowadays, the one area where you may encounter problems during installation is when it comes to custom hardware – particularly network-related, especially Wi-Fi! It's not at all uncommon to boot a live CD and find you have no connectivity. The first step to resolving the issues is finding out which hardware you have in your machine – use:

```
hwinfo --wlan -short
```

This will tell you which chipset your machine is running, which will allow you to look for the correct driver. The Linux Wireless page on kernel.org lists kernel-supported devices – **wireless.wiki.kernel.org/en/users/drivers** is a good starting point.

5 I CAN'T USE MY PRINTER

Although most of the large printer companies have some degree of Linux support, it's fair to say that support is pretty patchy but improving all the time. If you are having issues with your printer, or you are planning to buy a new printer, Ubuntu hosts a useful page detailing supported models – **bit.ly/2me87bG** – as does **LinuxPrinting.org**. Printing on Linux is enabled via the Common UNIX Printing System (CUPS), which is an open source, portable printing framework. After installing CUPS (with `apt install cups`), point your web browser to **http://localhost:631/admin** to access GUI-based configuration of the printing system.

6 UBUNTU INCORRECTLY SAYS MY LTS RELEASE IS THE LATEST

If you have installed a long-term support (LTS) version of Ubuntu, you may note that even though a new release is available, your system isn't showing it as available. This is because LTS releases typically prompt only to update to newer LTS releases. Of course, we generally want to be on the latest version regardless. So, to change this behaviour in Ubuntu, open Software and Updates, select the Updates tab and in the 'Notify me of new updates' section, change the option from 'For long-term support versions' to 'For any new version'. Check for an update in the usual way and you will be able to upgrade. To achieve the same result on the command line, edit **/etc/update-manager/release-upgrades** and ensure `Prompt=normal` is set. Use:

```
do-release-upgrade
```

...to start the upgrade.

7 I CAN'T INSTALL SECURITY UPDATES AUTOMATICALLY

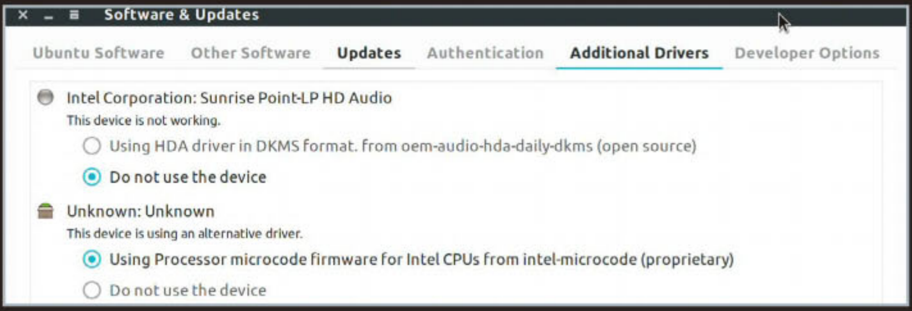
Out of the box, Ubuntu is configured so that all updates need to be user-installed. This is great if we want to have ultimate control and awareness of what our system is doing. However, it's probably not a bad idea to have our machine install security updates automatically at the very least. Configuring this is actually pretty straightforward – run the command:

```
sudo dpkg-reconfigure --priority=low unattended-upgrade
```

...then accept the prompts and you're all set. Note that reboots won't happen automatically; to enable this, `Unattended-Upgrade::Automatic-Reboot "true"` must be set in **/etc/apt/apt.conf.d/20auto-upgrades**.

8 I WANT TO INSTALL PROPRIETARY DRIVERS

In a default installation of Ubuntu, proprietary drivers are disabled. What are proprietary drivers? They are drivers that don't conform to the completely open source licences of the OS itself. Provided you are comfortable with this, installing these drivers for wireless cards, graphics cards and so on can provide big performance boosts or fix things that are otherwise broken. To enable, go to the Additional Drivers section of Software Sources, choose the drivers you wish to use and click Apply Changes. The selected drivers will then install.

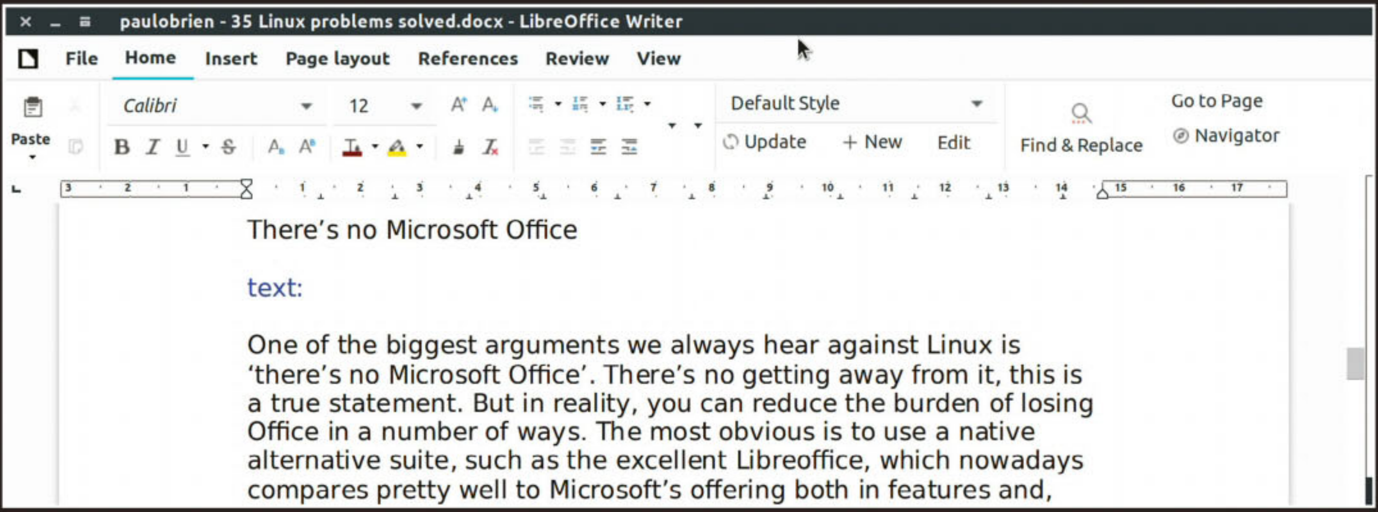


Intel Corporation: Sunrise Point-LP HD Audio
This device is not working.
○ Using HDA driver in DKMS format. from oem-audio-hda-daily-dkms (open source)
● Do not use the device
Unknown: Unknown
This device is using an alternative driver.
● Using Processor microcode firmware for Intel CPUs from intel-microcode (proprietary)
○ Do not use the device

Above Installing proprietary drivers can get hardware working when open source options aren't available

THIRD-PARTY APPLICATIONS

After your initial Linux install, you'll want to get the best third-party applications on your system right away



There's no Microsoft Office

text:

One of the biggest arguments we always hear against Linux is 'there's no Microsoft Office'. There's no getting away from it, this is a true statement. But in reality, you can reduce the burden of losing Office in a number of ways. The most obvious is to use a native alternative suite, such as the excellent Libreoffice, which nowadays compares pretty well to Microsoft's offering both in features and,

Above LibreOffice is a capable alternative to Microsoft Office

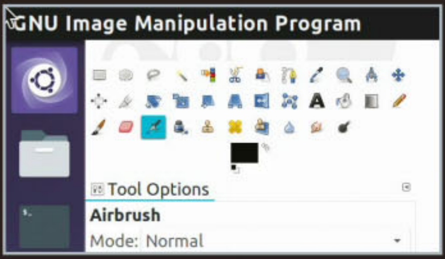
9 I NEED OFFICE!

One of the biggest arguments we always hear against Linux is 'there's no Microsoft Office'. There's no getting away from it: this is a true statement. But in reality, you can reduce the burden of losing Office in a number of ways. The most obvious is to use a native alternative suite, such as the excellent LibreOffice, which nowadays compares pretty well to Microsoft's offering both in features and, with the latest 'ribbon' user interface in LibreOffice 5.3, style too. LibreOffice includes alternatives to all the main Microsoft Office apps – Word (Write), Excel (Calc), PowerPoint (Impress), Visio (Draw) and even Access (Base).

Although this is definitely the purest option, even the most hardened Linux user will probably admit that there are times when only the real Office experience will do – sharing files with others is a key example of this. WINE 2.0 (or the Crossover or PlayOnLinux tools based on it) allows Windows applications to run fully on Linux, now with full support for Office 2013. Simply install the host app and run the Office installer inside it. It works surprisingly well. If neither of these works for you, you could consider running a bare Windows installation inside a VirtualBox or VMware virtual machine and installing into that. Although perhaps a little bit heavyweight, it works… and that's the most important thing!

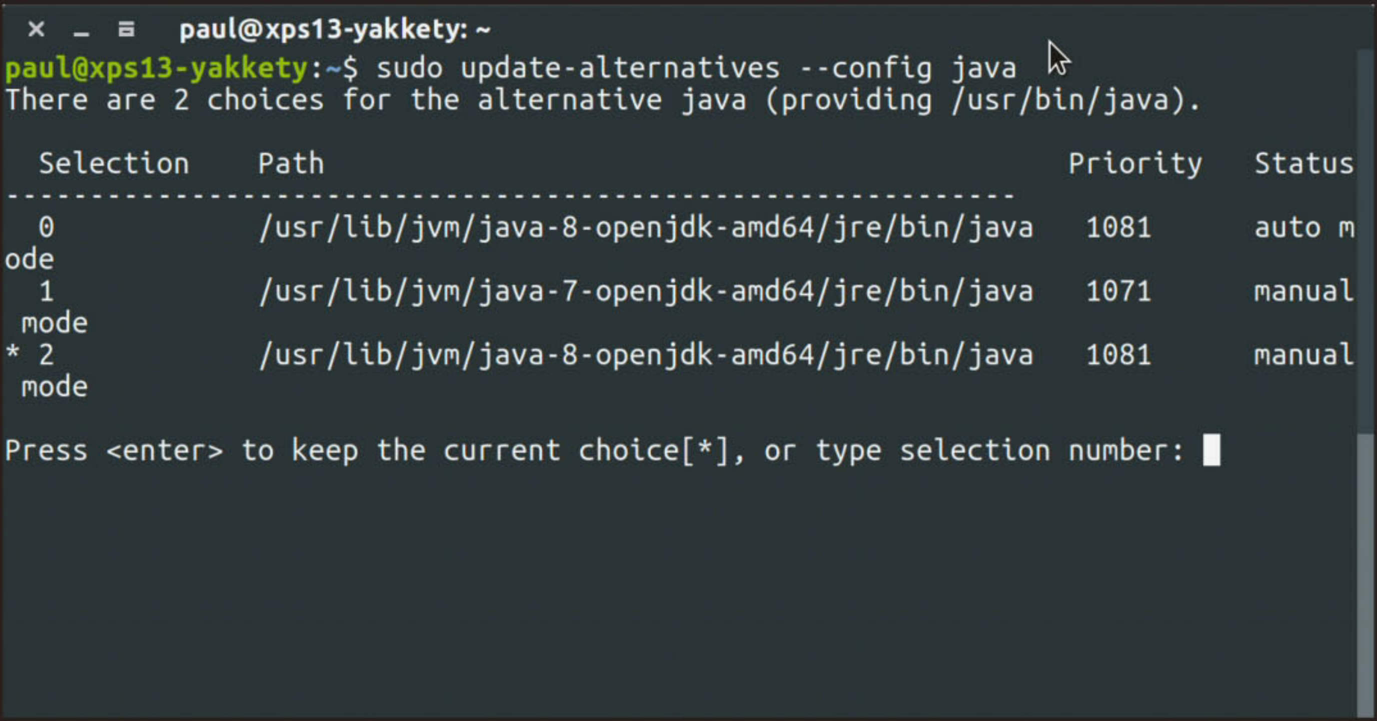
10 EVERYTHING IS TINY ON MY HIDPI DISPLAY

So you've got a machine with a fantastic new HiDPI display… perhaps an XPS 13 Developer Edition (UHD) or just a lovely new 4K monitor for your desktop. That's all great, except everything is tiny in the OS and some of your favourite apps. First up, don't be alarmed – this isn't a problem that's unique to Linux; even now, Windows 10 has some pretty nasty-looking UI elements on high-resolution screens. Fixing up the main OS itself is easy enough: you just need to set a menu and title bar scaling factor in the Screen Display section of Settings. After that's done, you'll need to look for specific apps that are problematic. Many will have their own HiDPI themes; however, you can also check out the window scaling options in gnome-tweak-tool to help you get everything scaled just right.



11 I REALLY NEED PHOTOSHOP

You've got your system up and running and you've realised that you don't need Microsoft Office any more thanks to LibreOffice. What's another big app that new Linux adopters struggle to come to terms with being without? Adobe Photoshop. There's no denying that Photoshop is an awesome application, but unless you really use a large number of the advanced features in Photoshop, Linux has a heavyweight image editor of its own that you can use – GIMP. Short for the GNU Image Manipulation Program, the application is an incredible tool for photo retouching, image composition and authoring, and much more thanks to its open source (and of course free) nature and extensible plug-in architecture. GIMP is also available on Windows and Mac OS X, and excels on those platforms too!



```
paul@xps13-yakkety:~$ sudo update-alternatives --config java
There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                          Priority   Status
------------------------------------------------------------
  0            /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java   1081      auto m
ode
  1            /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java   1071      manual
 mode
* 2            /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java   1081      manual
 mode

Press <enter> to keep the current choice[*], or type selection number: 
```

Above Linux doesn't include Java by default, but it can be installed in various ways

12 JAVA ISN'T INSTALLED

Out of the box, Linux doesn't include the Java programming language. When you do come to install it, there are a number of different options too. The first choice to make is whether to use open source (OpenJDK) or closed source (Oracle) Java. In reality, there isn't an awful lot between them nowadays; however, depending on what you are using Java for, you may need a specific version. OpenJDK is included in the repositories of most distributions and so can be installed with a command such as:

```
apt-get install openjdk-8-jre
```

To install Oracle's Java, the easiest way is to use a PPA, such as that hosted by webupd8team – use the commands:

```
sudo add-apt-repository ppa:
webupd8team/java
sudo apt-get update
sudo apt-get installer oracle-java8-
installer
```

Note that if you are not specifically doing Java development, you may only need the JRE (Java Runtime Environment) rather than the JDE (Java Development Kit) installed.

Should you need both types of Java installed, however, perhaps in multiple versions (Java 7 / 8 etc), you can do that – and switch between which one is enabled as the default on your system. Use the command:

```
sudo update-alternatives --config java
```

...to interactively choose your preferred version (and do the same for javac if required).

The default installation of Java won't enable support in your browser – you will need to manually install the plug-in if required. Oracle has a support page on its website detailing how to do so: **bit.ly/2lZUlh0.**

13 THIS APP HASN'T INSTALLED PROPERLY

We're willing to bet that if you've been using Linux for any time at all, you're already au fait with installing applications directly from the command line rather than using the GUI installer. Most new users learn about application dependencies quickly and rudely – trying to install the Chrome browser using the GUI quickly becomes an exercise in learning how to use the command line, for example. You'll notice that when you try to install DEB packages, sometimes they will fail to install because dependencies are missing and not automatically resolved. Should you experience this (as in the case of Chrome), you can normally automatically fix the issues using the command:

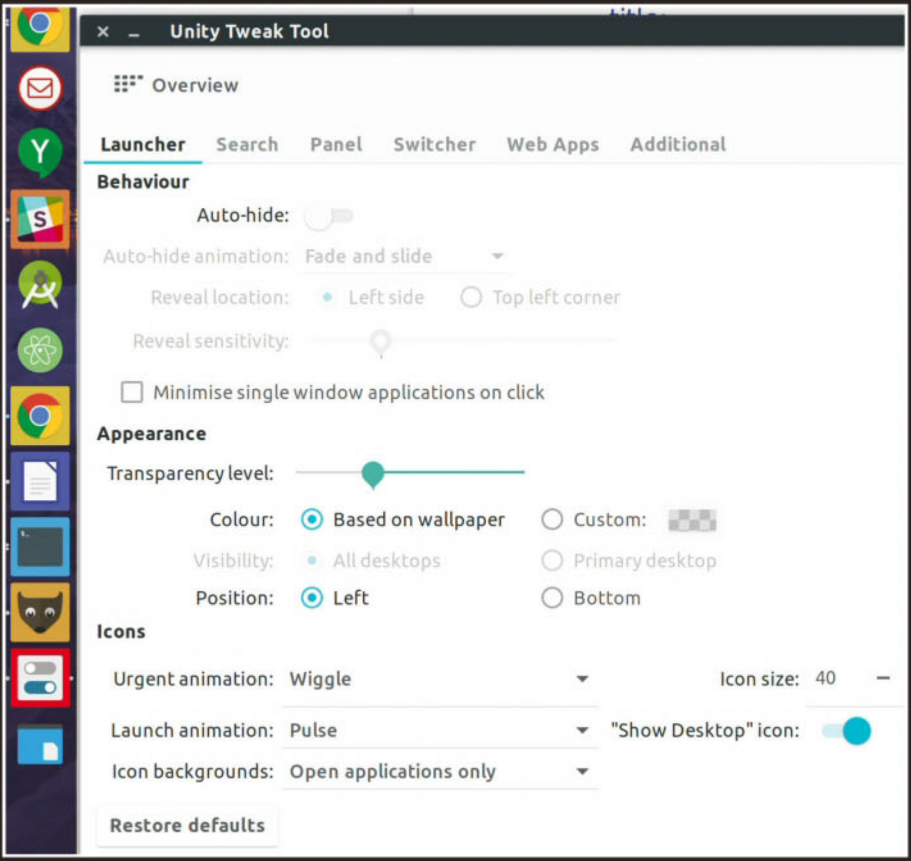
```
apt-get install -f
```

14 I CAN'T PLAY ANY VIDEOS

New arrivals on Linux are often surprised to find that there's not a capable media player pre-installed, so playing video files can be a problem. Fortunately, there's an easy solution in the form of VLC. VLC is a free and open source cross-platform multimedia player and framework that plays most multimedia files as well as DVDs, audio CDs, VCDs and various streaming protocols. To install on your Linux machine, either do so from your distribution's repositories (we guarantee it's there!) or using the packages available on **www.videolan.org.** A browser plug-in (browser-plugin-vlc) and additional streaming codecs (libavcodec-extra-53) are also available. In order to play region-locked DVDs, you will need to install libdvdcss2 using the guide on the VLC site.

TWEAKS AND IMPROVEMENTS

Linux is almost infinitely tunable to your specific needs, but there are a few key things you're going to want to do



Unity Tweak Tool

Overview

Launcher | Search | Panel | Switcher | Web Apps | Additional

Behaviour

Auto-hide:

Auto-hide animation: Fade and slide

Reveal location: ● Left side ○ Top left corner

Reveal sensitivity:

□ Minimise single window applications on click

Appearance

Transparency level:

Colour: ◉ Based on wallpaper ○ Custom:

Visibility: ● All desktops ○ Primary desktop

Position: ◉ Left ○ Bottom

Icons

Urgent animation: Wiggle

Icon size: 40

Launch animation: Pulse

"Show Desktop" icon:

Icon backgrounds: Open applications only

Restore defaults

Above The Unity tweak tool provides an easy way to toggle some normally hidden Unity settings

15 BATTERY LIFE IS WORSE THAN WINDOWS

It's a sad reality that battery life on a portable device running Linux is often worse than the same machine running on Windows. This is largely down to the fact the manufacturer will have made specific battery optimisations for the device / OS combo, the drivers being more mature. One secret weapon to help you mitigate this situation, however, is TLP. TLP brings you the benefits of advanced power management for Linux without the need to understand every technical detail. TLP comes with a default configuration that is already optimised for battery life, so you can just install and forget, although it is highly customisable to fulfil your specific requirements if preferred. After installation (TLP is in most distribution repos), it can be started with:

```
sudo tlp start
```

Then use:

```
sudo tlp-stat -s
```

...to check the system is working properly.

16 I WANT A 'DESKTOP' SHORTCUT

Sometimes it's the little things that make the difference and this is a great example – you won't believe how useful it is to have a 'Desktop' shortcut on your launcher. Even better, it's typically nice and simple to enable. The exact steps depend on the distribution you are using, but using Ubuntu with Unity as an example, simply install the unity-tweak-tool, navigate to the Launcher section and toggle the Show Desktop icon setting. The icon should appear straight away. The app has a host of other useful tweaks too!

17 I'M MISSING SOME IMPORTANT FONTS

If you are loading documents you've received from others, you may notice you're missing some fonts that are installed by default on a Windows PC. Luckily, there's an easy fix. Simply install the ttf-mscorefonts-installer package and your system will be updated with Andale Mono, Arial, Comic Sans MS (yuck!), Courier New, Georgia, Impact, Times New Roman, Trebuchet, Verdana and Webdings. As this is a basic set of fonts, you will still be missing some Cleartype fonts such as Calibri and Cambria. Unfortunately, these are not officially available for Linux.

18 THE STOCK THEME IS KINDA UGLY

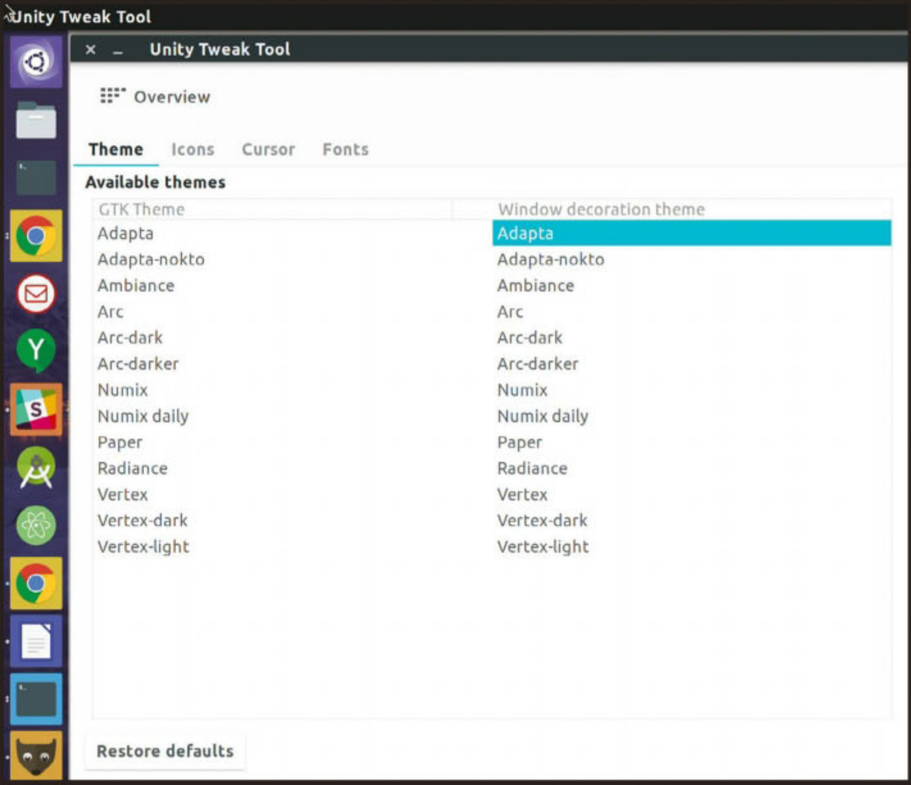
Although things are much better than they used to be, Linux's graphical interfaces still aren't the most beautiful things out of the box, even on the Ubuntu flagship distribution running Unity. That's the bad news – the good news is that most distros are easily themable now. Try this as a nice simple way to make your Ubuntu machine look a lot slicker than before. Install unity-tweak-tool. Download the 'Adapta' theme from **github.com/adapta-project/adapta-gtk-theme**. Then download the paper icons from **snwh.org/paper**. Use the tweak tool to enable both. Now, doesn't that look like a whole different OS? Beautiful.

19 I CAN'T WORK GRUB ON MY TABLET

There are a lot more convertible laptops running with powerful hardware now, which means they also make excellent Ubuntu machines (the Dell XPS / Latitude 12 7275 are perfect examples). There's a good chance that you'll dual-boot your machine, which can immediately throw up an issue. GRUB is not exactly what we'd call touch-optimised, is it? With the default config, GRUB will just sit waiting for input indefinitely. Not ideal. To get around this problem, you should update the GRUB config to add a default option and a timeout. Edit **/etc/default/grub**, setting GRUB_DEFAULT as desired (the list is 0 based) and an appropriate GRUB_TIMEOUT value in seconds. Also, set GRUB_HIDDEN_TIMEOUT_QUIET to false. After running:

```
sudo update-grub
```

...the problem is solved. You can now at least boot your machine without a keyboard!



GTK Theme	Window decoration theme
Adapta	Adapta
Adapta-nokto	Adapta-nokto
Ambiance	Ambiance
Arc	Arc
Arc-dark	Arc-dark
Arc-darker	Arc-darker
Numix	Numix
Numix daily	Numix daily
Paper	Paper
Radiance	Radiance
Vertex	Vertex
Vertex-dark	Vertex-dark
Vertex-light	Vertex-light

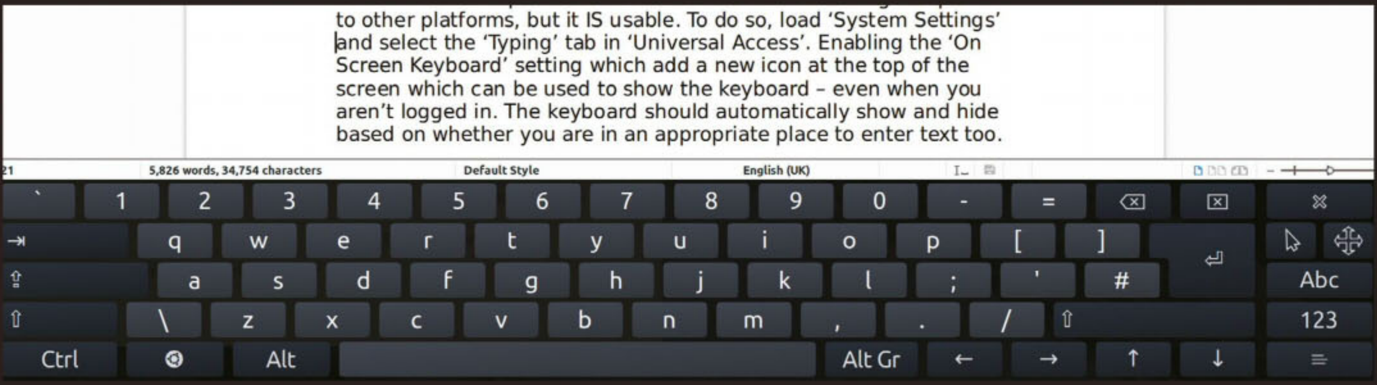
Above Themes, custom icons, custom cursors and font settings can transform the appearance of your Linux install

20 THERE'S NO ON-SCREEN KEYBOARD AVAILABLE

Once you've managed to get into Ubuntu on your tablet, you'll come across a new problem – no on-screen keyboard. It's fair to say that the tablet experience on Linux is somewhat lacking compared to other platforms, but it is usable. To do so, load System Settings and select the Typing tab in Universal Access. Enabling the On Screen Keyboard setting will add a new icon at the top of the screen, which can be used to show the keyboard – even when you aren't logged in. The keyboard should automatically show and hide based on whether you are in an appropriate place to enter text, too.

21 I WANT TO INSTALL SOME ANTIVIRUS

"Only Windows needs antivirus". Famous last words? Although Windows does suffer more attacks than either Linux or Mac OS X, simply due to the size of its install base, it's good practice to have antivirus installed if you are a Linux user too. Most of the large antivirus companies offer protection – and often for free. The most widely used antivirus on Linux is ClamAV, which doesn't come with a GUI by default, but does have a third-party GUI offering in the form of the lightweight clamtk front-end, which is available from **dave-theunsub.github.io/clamtk**.



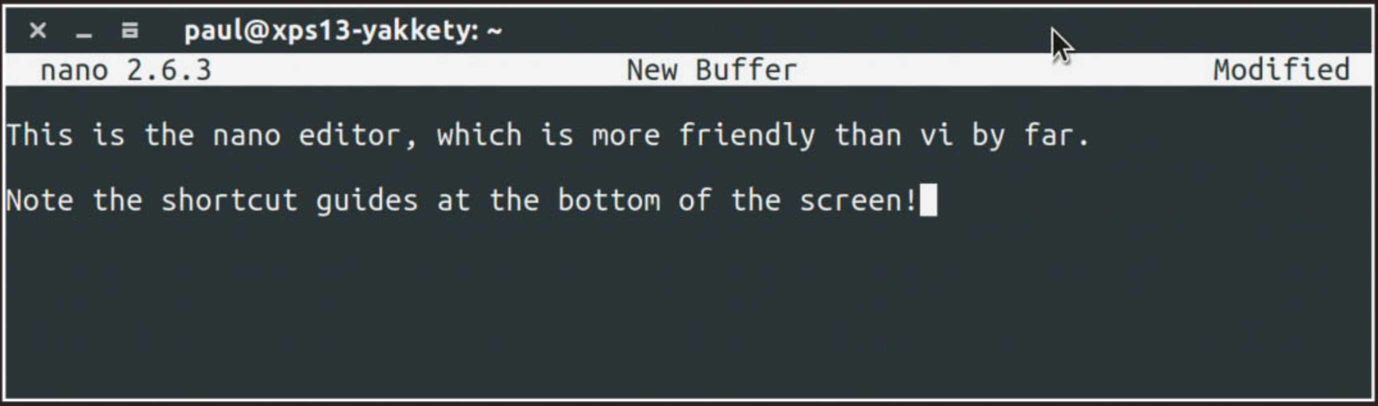
to other platforms, but it IS usable. To do so, load 'System Settings' and select the 'Typing' tab in 'Universal Access'. Enabling the 'On Screen Keyboard' setting which add a new icon at the top of the screen which can be used to show the keyboard - even when you aren't logged in. The keyboard should automatically show and hide based on whether you are in an appropriate place to enter text too.

21 | 5,826 words, 34,754 characters | Default Style | English (UK)

Above The Ubuntu on-screen keyboard allows you to use the OS on a tablet device

THE COMMAND LINE

At some point even the most hardened GUI fan has to delve into the command line, which can lead to problems unless you're prepared…



```
  nano 2.6.3                      New Buffer                      Modified

This is the nano editor, which is more friendly than vi by far.

Note the shortcut guides at the bottom of the screen!
```

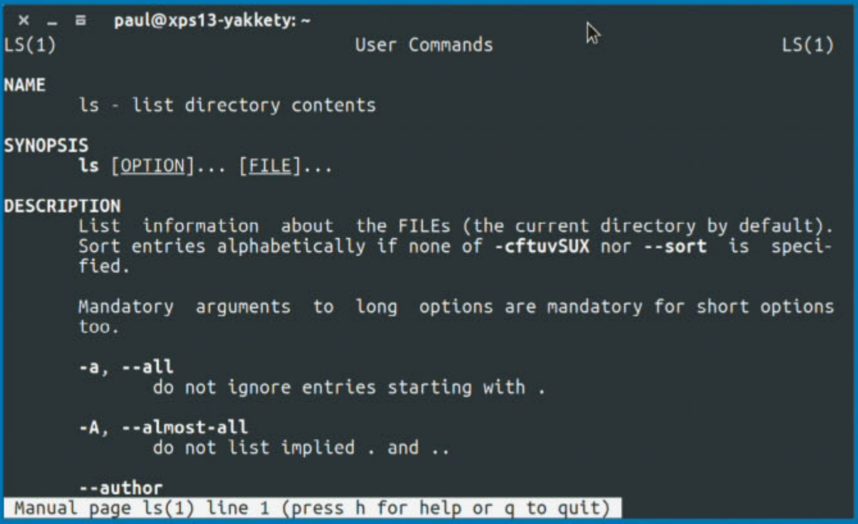
```
This is the gedit editor. If you really want a proper GUI, this is an option!
```

22 I FIND VI TOO HARD TO USE

When using the command line on your Linux machine, the commonly recommended editor is vi – with good reason, since it's a very powerful tool. What it isn't, however, is beginner-friendly – the commands can be pretty daunting. A solid alternative to vi is nano. This is an editor that still runs in the terminal, but is a little bit more accessible and holds the user's hand just a little more without being cumbersome. Of course, if you want to edit from the command line but really want to do so in a GUI, you can use gedit or a more advanced GUI editor such as Atom from **atom.io**.

23 I WANT TO SAFELY DO THINGS AS ROOT

An easy trap to fall into as a new Linux user is to do too much as root. Switching to the root user is very easy, but once you've got into the habit, it's a hard one to get out of. Here's a tip – change the root password to something you won't remember without using a password manager or looking it up, then try to use sudo instead (which runs under your account but elevates your permissions).



```
LS(1)                        User Commands                        LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about  the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort  is  speci-
       fied.

       Mandatory  arguments  to  long  options are mandatory for short options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
 Manual page ls(1) line 1 (press h for help or q to quit)
```

24 I OFTEN FORGET COMMAND-LINE OPTIONS

Linux has a dizzying array of command-line tools, each with its own nuanced syntax, which means that if you aren't using them day in, day out, remembering exactly how to use them can be a struggle. Fortunately, there are standard ways of finding out the correct commands. The first stop for help is using the --help suffix on a command (eg `ls --help`). This will print a short list of all the possible arguments, as well as (typically) an example and a weblink to full documentation. Should this scroll past too quickly for you to read, pipe to the more binary – eg `ls --help|more` – and it will be displayed a screen at a time; press the space bar to continue.

For a more detailed explanation of how a command is used, use the man function (eg `man ls`). This opens the manual page for the command, which you can again scroll through using the space bar.

25 I WANT TO KNOW WHAT SOFTWARE IS INSTALLED

With a number of different ways to install software on Linux, the best way to view what's installed is not often clear. If you are manually compiling and installing software, then it is particularly hard to keep track. As a rule, try to install from packages when they are available.

For a graphical view of what's installed in Ubuntu, use the Ubuntu Software application and click the Installed tab (this will also show Updates on a tab). Unfortunately, this will only show you what is installed using the GUI, so it is of limited use. Command-line alternatives are:

```
dpkg -getselections
```

Or…

```
apt-show-versions
```

Of course, if you are looking for something specific you can pipe through grep. For example:

```
dpkg --getselections|grep php
```

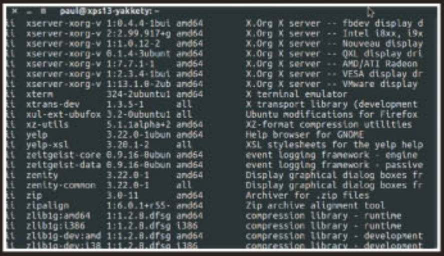
To see specifically which files are installed by a package, use:

```
dpkg -L packagename
```

If you'd like to know a little bit more about what each package is, you can get a list with a short description using:

```
dpkg-query -l
```

If you are manually compiling and installing from source, is there any way to keep track? Typically, a good approach is to keep the source for any apps you compile in a common directory that you can refer to later (even if you compress the source after installing). This will also give you the ability to use make `uninstall` to remove the software later.



```
paul@xps13-yakkety: ~
ii  xserver-xorg-v 1:0.4.4-1bui amd64        X.Org X server -- fbdev display d
ii  xserver-xorg-v 2:2.99.917+g amd64        X.Org X server -- Intel i8xx, i9x
ii  xserver-xorg-v 1:1.0.12-2   amd64        X.Org X server -- Nouveau display
ii  xserver-xorg-v 0.1.4-3ubunt amd64        X.Org X server -- QXL display dri
ii  xserver-xorg-v 1:7.7.1-1    amd64        X.Org X server -- AMD/ATI Radeon
ii  xserver-xorg-v 1:2.3.4-1bui amd64        X.Org X server -- VESA display dr
ii  xserver-xorg-v 1:13.1.0-2ub amd64        X.Org X server -- VMware display
ii  xterm          324-2ubuntu1 amd64        X terminal emulator
ii  xtrans-dev     1.3.5-1      all          X transport library (development
ii  xul-ext-ubufox 3.2-0ubuntu1 all          Ubuntu modifications for Firefox
ii  xz-utils       5.1.1alpha+2 amd64        XZ-format compression utilities
ii  yelp           3.22.0-1ubun amd64        Help browser for GNOME
ii  yelp-xsl       3.20.1-2     all          XSL stylesheets for the yelp help
ii  zeitgeist-core 0.9.16-0ubun amd64        event logging framework - engine
ii  zeitgeist-data 0.9.16-0ubun amd64        event logging framework - passive
ii  zenity         3.22.0-1     amd64        Display graphical dialog boxes fr
ii  zenity-common  3.22.0-1     all          Display graphical dialog boxes fr
ii  zip            3.0-11       amd64        Archiver for .zip files
ii  zipalign       1:6.0.1+r55- amd64        Zip archive alignment tool
ii  zlib1g:amd64   1:1.2.8.dfsg amd64        compression library - runtime
ii  zlib1g:i386    1:1.2.8.dfsg i386         compression library - runtime
ii  zlib1g-dev:amd 1:1.2.8.dfsg amd64        compression library - development
ii  zlib1g-dev:i38 1:1.2.8.dfsg i386         compression library - development
```

26 THE LOCATE COMMAND DOESN'T WORK

The 'locate' command is the quickest and easiest way to find files on Linux. Simply use it by typing `locate filename` and the system will return all instances of 'filename' (wildcards and regular expressions are supported) to which the requesting user has access. Adding the `-i` switch makes the search case-insensitive.

What if the locate command isn't bringing up files that you know are there? The database is updated periodically (usually in the daily cron), which means recently created files and directories won't show up immediately. To update the database manually, simply use:

```
sudo updatedb
```

> "You start a new session and you have no idea if the previous operation completed. Nightmare!"

27 WHEN SSH DISCONNECTS, I GET LOST

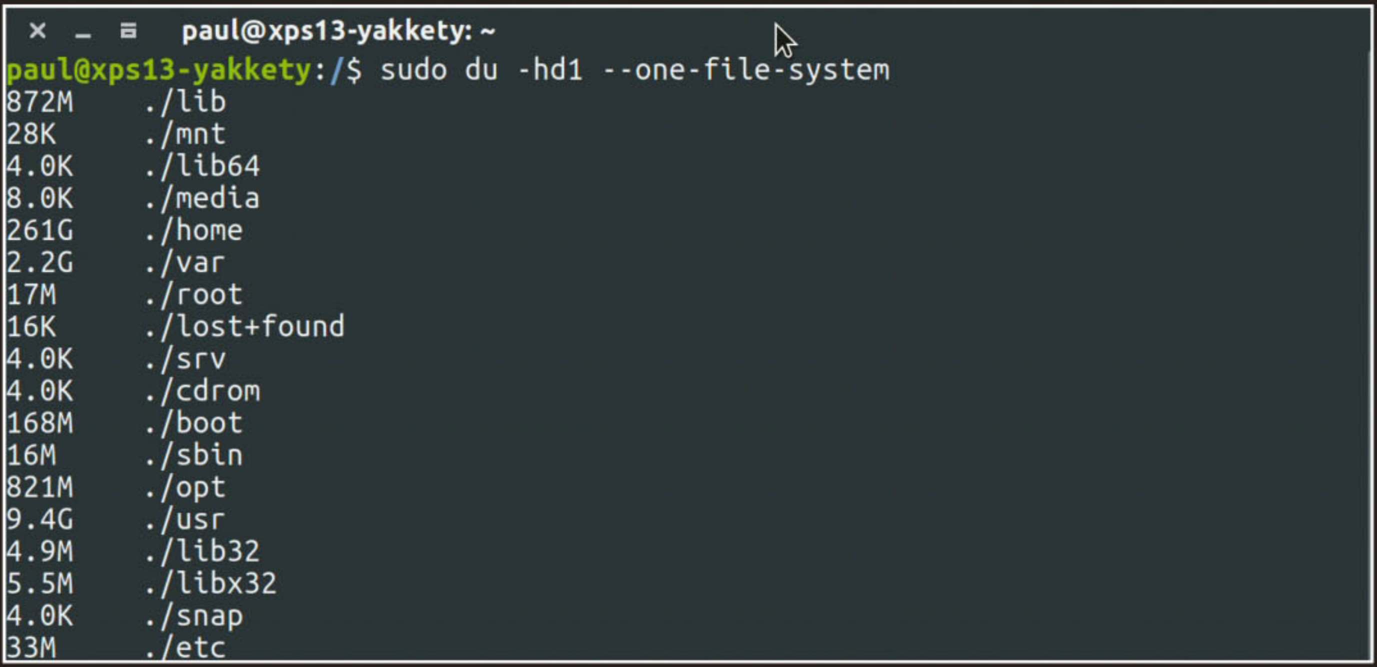
Picture the scene. You're SSHed into a remote server, doing something really important. You're just in the middle of running a command on the database when your Wi-Fi drops. You connect back to the server immediately, but you start a new session and you have no idea if the previous operation completed. Nightmare!

A great solution to this is the 'screen' tool. This is a full-screen window manager that multiplexes a physical terminal between several processes, typically interactive shells. Each virtual terminal has a separate scrollback history buffer and a copy-and-paste mechanism that allows the user to move text regions between windows. When screen is first opened, it creates a single window with a shell in it (or the specified command) and then gets out of your way so that you can use the program as you normally would. Then, at any time, you can create new (full-screen) windows with other programs in them (including more shells), kill the current window, view a list of the active windows, switch between windows, and more. All windows run their programs completely independent of each other and, most importantly, programs continue to run when their window is currently not visible and even when the whole screen session is detached from the user's terminal.

After installing screen, it's a good idea to just add it to your .profile so it's started on a new connection. Use `screen -RD` to reattach to a session else start a new one, and detach a remote session if needed.

28 OOPS, I USED CAT ON A BINARY FILE

This is a classic problem that everyone does at one time or another. You intend to cat a text file, but instead you accidentally cat a binary file. Your display goes mad; the computer is beeping at you like crazy, so you hit **Ctrl+C** like a madman. Nothing happens, so you keep hitting **Break** and finally, thank heaven, the screen stops scrolling and you are back at your command prompt. Hurrah! Only… everything is not right. The character set has gone completely haywire and what you are typing definitely doesn't match what's on the screen. Not good. If only there was any easy way to get things back to normal! Fortunately… there is. Simply run reset and your terminal will be un-binaried. Phew!



```
paul@xps13-yakkety: ~
paul@xps13-yakkety:/$ sudo du -hd1 --one-file-system
872M    ./lib
28K     ./mnt
4.0K    ./lib64
8.0K    ./media
261G    ./home
2.2G    ./var
17M     ./root
16K     ./lost+found
4.0K    ./srv
4.0K    ./cdrom
168M    ./boot
16M     ./sbin
821M    ./opt
9.4G    ./usr
4.9M    ./lib32
5.5M    ./libx32
4.0K    ./snap
33M     ./etc
```

Above 'du' is the perfect tool for spotting those rogue huge directories

LINUX SERVERS

If you are deploying Linux to a server, then you can expect to experience a somewhat different set of challenges

29 I'M RUNNING OUT OF DISK SPACE

When deploying Linux to VPS servers or containers, we seem to have gone back in time somewhat – we're again in a world of limited disk space in which every byte counts! It's ironic, but fortunately Linux has some tools to help us see where our valuable disk space is going. The first useful command-line tool is 'df', which shows us space on each volume and how much is used. Use df -h to get a more human-readable version. Next up is 'du', which summarises the disk usage of files / directories. Again, the du -h switch can be used to make the output more human-readable. Another particular switch is to use the command in the following way: du -hd1. This means to show the usage, in human-readable form, but to a max depth of one directory. So totals will be summed for every subdirectory, but presented as totals for paths one directory from the current one.

30 INSTALLED APPS AREN'T UPDATING

Earlier, we talked about configuring Linux to apply automatic security updates, but it's worth remembering that application updates themselves and less critical updates won't be installed. Generally speaking, automatically updating everything isn't the best idea in the world, but if you want to do it, you can! In a similar way to the previous example, you can use:

```
sudo apt install unattended-upgrades
```

...in order to install the appropriate file to **/etc/apt/apt.conf.d/50unattended-upgrades** and uncheck the appropriate line to allow all types of updates.

A useful package to install is apticron, which is used to email an administrator info about any packages on the system that have updates available, plus a summary of changes in each package. After installation, set the destination email address in **/etc/apticron/apticron.conf**.

31 MY SERVER IS BOOTING TO X

If you are going to run Linux on a server, then of course the best approach is to install a server-specific distribution. Sometimes, of course, this isn't always practical – you may start with a desktop machine that becomes a headless server, or you might just prefer to use a GUI to get set up. Either way, if you're not using the graphical interface, you probably don't want it starting up on your machine. To boot to text mode on each boot, edit **/etc/default/grub**, comment out the GRUB_CMDLINE_LINUX_DEFAULT line and ensure GRUB_CMDLINE_LINUX="text" is set. After running:

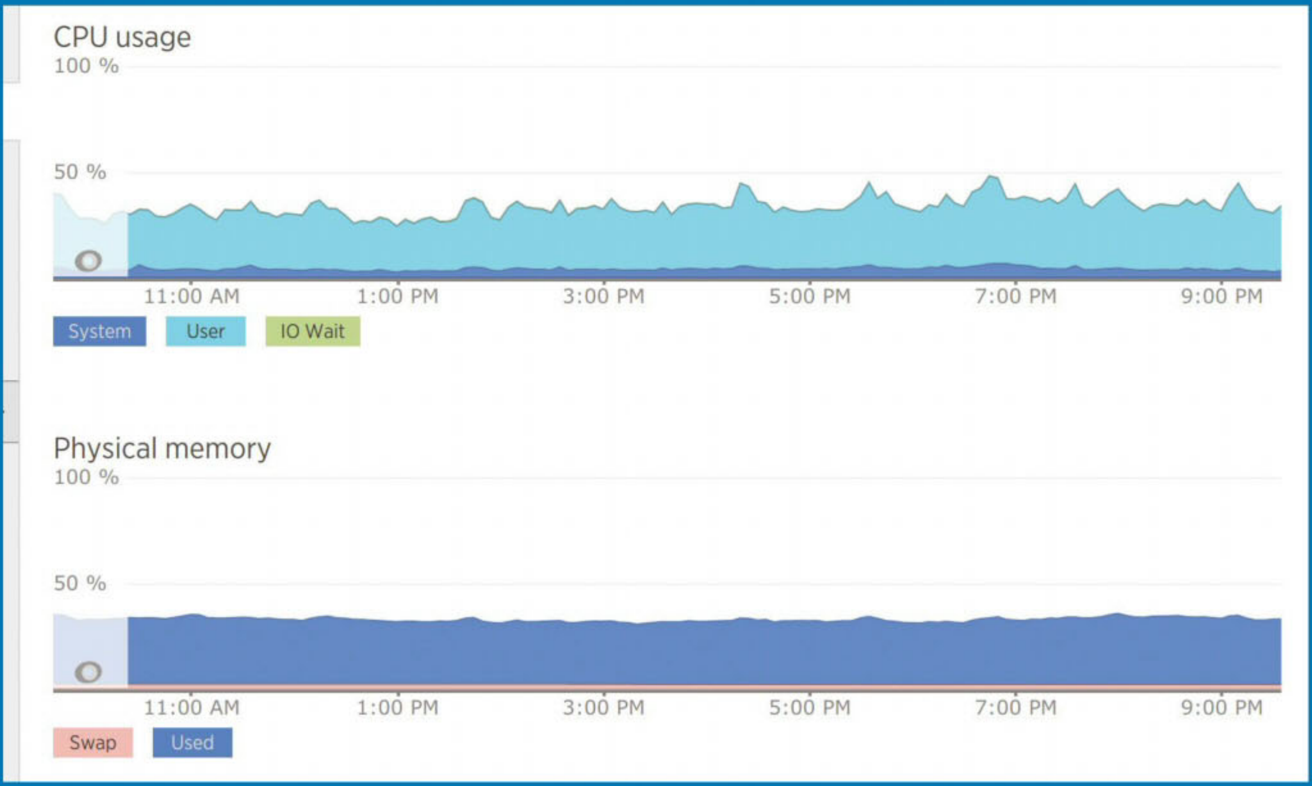
```
sudo update-grub
```

...you're all set. If running a systemd-enabled system (Debian 8/Ubuntu 15.04 onwards), you'll also need to run:

```
sudo systemctl set-default multi-user.target
```

Undo by running:

```
sudo systemctl set-default graphical.target
```



32 THE CPU AND RAM USAGE IS TOO HIGH

If you're running a server, it makes sense to keep an eye on CPU and RAM usage so you know that your system is coping with the load you are putting on it. A good way to do this is using third-party services such as New Relic; however, on the box itself you can also check usage using the 'top' commands. Running top on its own will sort by top processor users; however, if you hit capital M (**Shift+M**), sorting will be by memory usage. Remember that high CPU and particularly memory usage isn't necessarily a bad thing, especially on a server running fairly consistent processes. Unused memory is wasted memory!

33 PHP ISN'T WORKING IN MY LAMP STACK

When configuring the LAMP (Linux, Apache, MySQL, PHP) stack on your server, there is a simple trap you can fall into. After installing the PHP packages, you may find that your scripts aren't being executed and are instead being printed to the screen, despite restarting Apache2 and doing everything you think you should be doing. The usual culprit? Forgetting to install the libapache2-mod-php package, which integrates PHP with the web server. It's a common mistake, easily done, and it's not the easiest package name to remember off the top of your head!

> “You will need to install an MTA”

34 MY SERVER WON'T SEND OUTGOING MAIL

If you want to send email from your Linux server, you will need to install an MTA (mail transfer agent) such as Postfix in order to do so. The best way to install Postfix and associated utilities on Ubuntu is using:

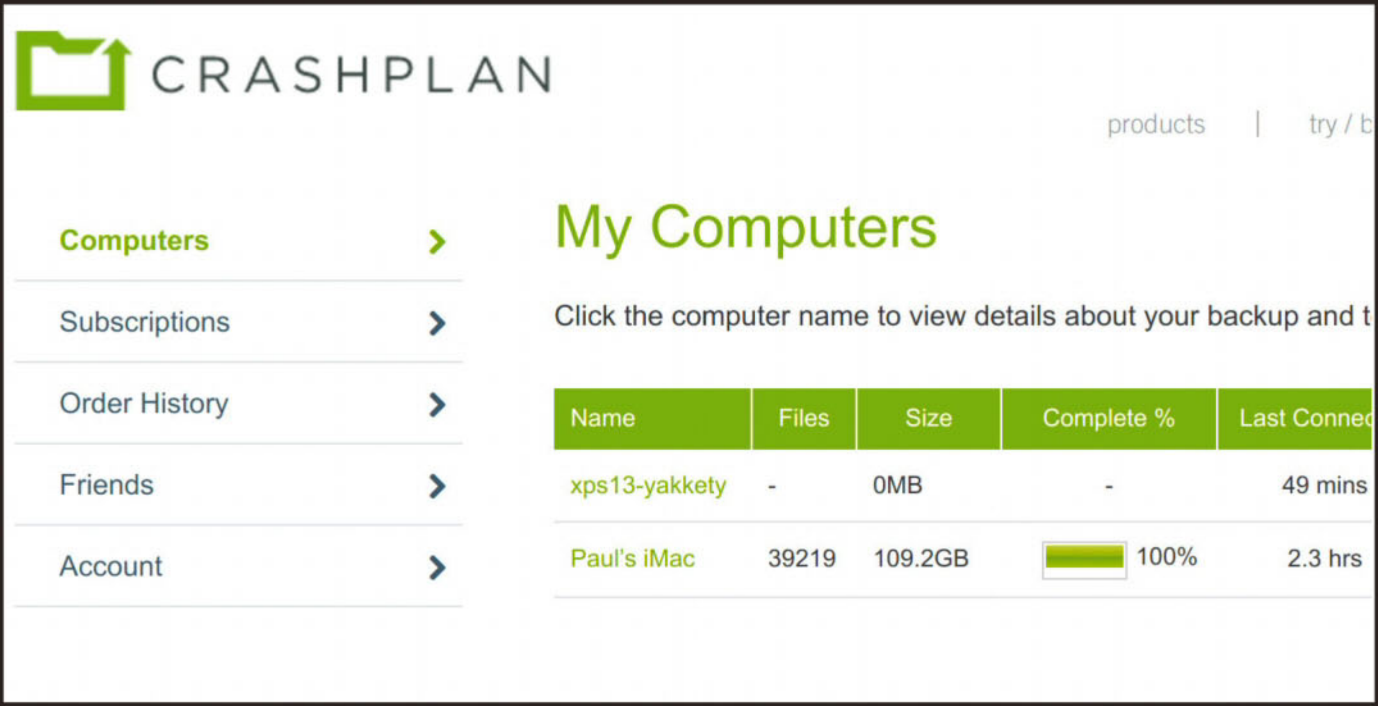
```
sudo apt-get install mailutils
```

Near the end of the installation process, you will be prompted to select a 'mail server configuration that best suits your needs' – choose 'Internet site' (the default option). You'll be prompted to provide a domain name; ensure the DNS for this name matches the IP address of your server. Now the most important step – to set the server to only allow mail sending from localhost, edit **/etc/postfix/main.cf** and replace `inet_interfaces = all` with `inet_interfaces = localhost`. To test sending mail, use the command:

```
echo "This is a test email" | mail
-s "This is a test subject" me@
myemailaddress.com
```

JUST IN CASE...

What happens if it all goes wrong? You may have a hardware issue or your Linux problem means you need to wipe your disk



My Computers

Click the computer name to view details about your backup and t

Name	Files	Size	Complete %	Last Connec
xps13-yakkety	-	0MB	-	49 mins
Paul's iMac	39219	109.2GB	100%	2.3 hrs

Above A comprehensive, tested backup solution will give you peace of mind – particularly as you tinker around with your Linux install!

35 I NEED A BACKUP PLAN

You finally have your Linux machine exactly how you like it. Well done! So do you fancy going through that process again? We suspect not. If your hard disk fails, do you have vital data you want to save? We imagine so. With this in mind, you should ensure that you have a reliable backup strategy for your machine, whether it is a desktop or a server.

Commercial solutions are available such as Crashplan which are very accomplished (with free friend backup) and of course there are excellent free equivalents such as CloudBerry Backup and open source options too, such as Backula or Bareos.

If you want to have ultimate control over your backups, you may wish to roll your own. There are a multitude of different backup scripts available for download across the internet, which provide an ideal starting point for your own script. How it works will depend on what you want to back up – on a desktop it is likely to be primarily file-based, but on a server you may also want to back up databases. A common approach is to prepare a backup archive locally and store it remotely on a service such as Amazon S3, automatically deleting files after a specified amount of time.

Don't forget one of the golden rules of backup: ensure that you test your backups with a restore now and again. Admittedly, this is easier to do on servers than workstations, but if you never put your solution to the test, you can't be 100 per cent sure it works!

> “If you want to have ultimate control over your backups, you may wish to roll your own”

Did you know that European forests, which provide wood for making paper and many other products, have grown by 44,000km^2 over the past 10 years? That's more than 1,500 football pitches every day![†]

Love magazines? You'll love them even more knowing they're made from natural, renewable and recyclable wood

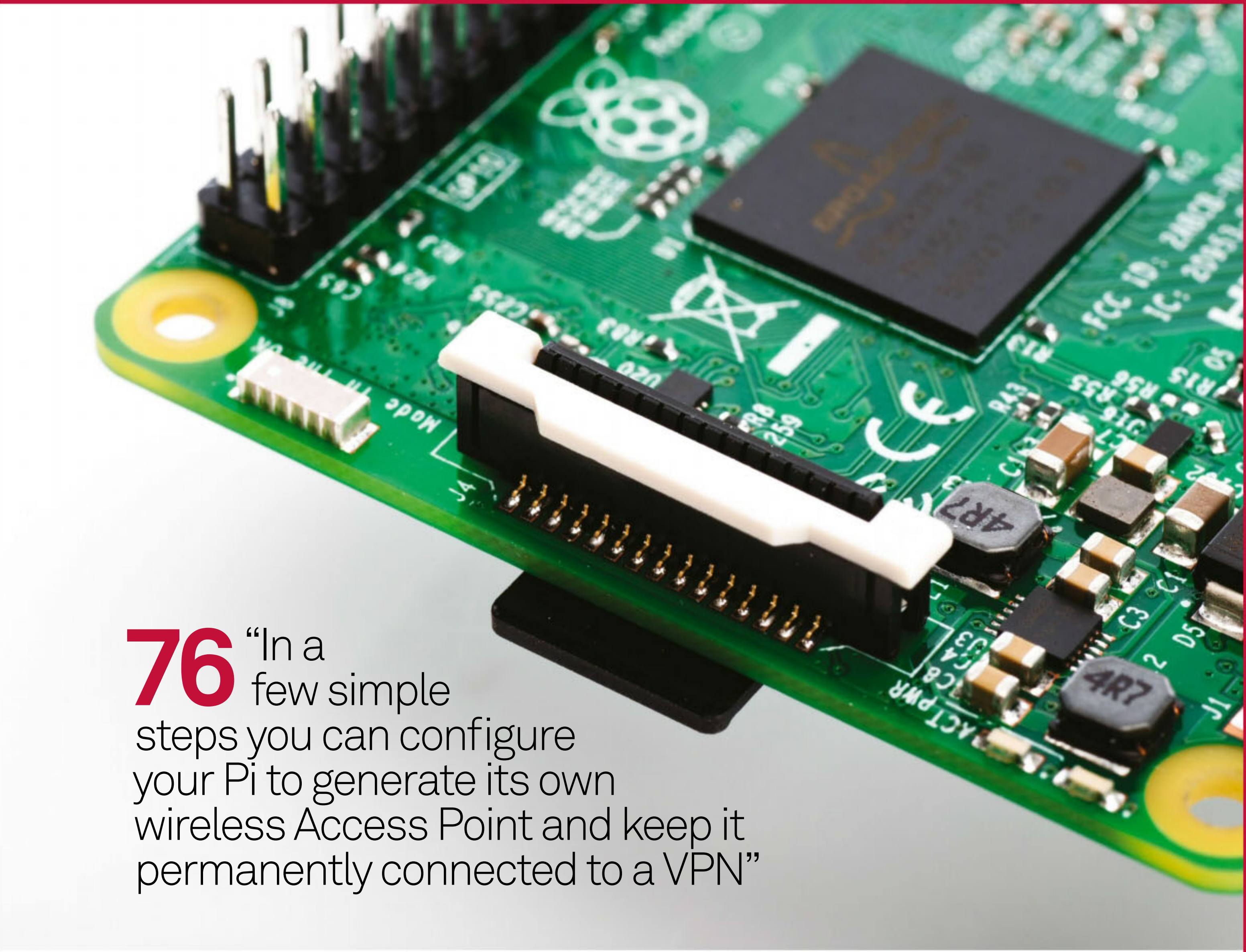
[†] UNFAO, Global Forest Resources Assessment 2005-2015.

Two Sides is a global initiative promoting the responsible use of print and paper which, when sourced from certified or sustainably managed forests, is a uniquely powerful and natural communications medium.

There are some great
reasons to #LovePaper
Discover them now,
twosides.info

TWO SIDES

PRACTICAL Raspberry Pi



76 "In a few simple steps you can configure your Pi to generate its own wireless Access Point and keep it permanently connected to a VPN"

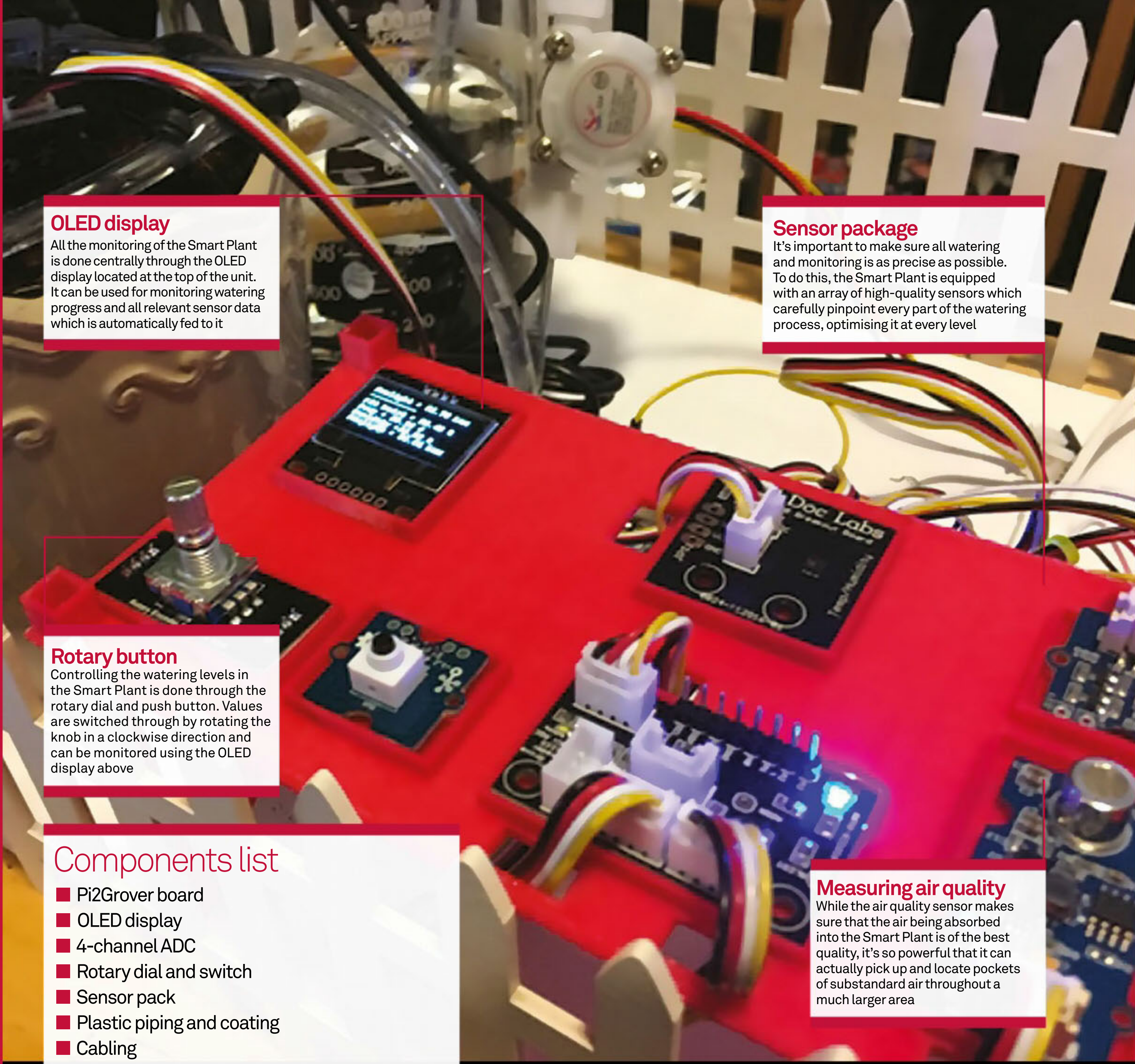
Contents

72 Automate your indoor garden

74 Get hands-on with the Pimoroni Blinkt!

76 Make a Raspberry Pi VPN Access Point

80 Anaconda for the Raspberry Pi



OLED display

All the monitoring of the Smart Plant is done centrally through the OLED display located at the top of the unit. It can be used for monitoring watering progress and all relevant sensor data which is automatically fed to it

Sensor package

It's important to make sure all watering and monitoring is as precise as possible. To do this, the Smart Plant is equipped with an array of high-quality sensors which carefully pinpoint every part of the watering process, optimising it at every level

Rotary button

Controlling the watering levels in the Smart Plant is done through the rotary dial and push button. Values are switched through by rotating the knob in a clockwise direction and can be monitored using the OLED display above

Measuring air quality

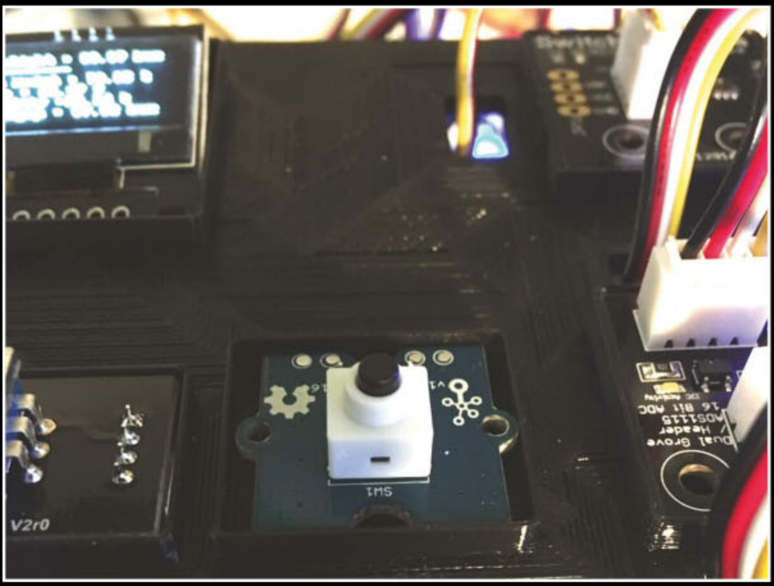
While the air quality sensor makes sure that the air being absorbed into the Smart Plant is of the best quality, it's so powerful that it can actually pick up and locate pockets of substandard air throughout a much larger area

Components list

- Pi2Grover board
- OLED display
- 4-channel ADC
- Rotary dial and switch
- Sensor pack
- Plastic piping and coating
- Cabling
- USB power control
- USB controlled pump

Above Amazon Echo owners can connect their unit to the Smart Plant and monitor progress using just their voice. A full setup tutorial will be available from **switchdoc.com** soon

Right The Grove push button is used as the central control for the SmartPlant. It works in tandem with the Pi for all core features and functions



Automate your indoor garden

John Shovic's indoor garden monitoring system is the perfect project for new and seasoned Pi makers alike

We've seen some pretty unique projects using the Pi, but the SmartPlant is entirely different. How did the concept come about?

The original SmartPlantPi idea came from where many good ideas come from: a discussion in a local pub. We were brainstorming about what we could build with the amazing amount of environmental sensors out there (while staring at my beer) and I suggested that we build an automated plant-watering system for the Raspberry Pi. What made me think of that? I had been teaching a class at the local university, CS270 Systems Programming, in which I had Raspberry Pi computers for each student to take home and build projects.

Linux can be an intimidating study when you are down in the command line studying the kernel and the multitasking system. I decided to have the students do what I call 'Physical Computing', which is where we build sensor-based projects, like a sunlight analyser, and then use LEDs and other devices for the Raspberry Pi to interact with the environment.

How long did it take you to go from concept to a fully made product? Did you encounter any issues during the build process?

The development of the SmartPlantPi kit took about three months from [the] start to the finish of the project and kit documentation. The big choice we made in the hardware design was the decision to go with the Grove connector system. The Grove system is a standard plug used to connect a variety of small computer devices together. All told, there are about 200 different Grove sensors out there, from a number of different manufacturers. All of the Grove connectors are the same size and have four wires but come in four flavours: I²C, Digital, Analog and Serial. Going with an all-Grove design means there is absolutely no soldering required for SmartPlantPi, which makes the project much more accessible for all levels of makers.

For those who may be unaware of the SmartPlantPi, could you give them an overview of what the project is?

SmartPlantPi (or Smart Plant for short) is an introductory, easy-to-build Raspberry Pi-based environmental monitoring and plant watering system using advanced sensors to monitor the soil moisture, monitor the sunlight, watch the air quality and monitor temperature and humidity. It is designed to be easily and simply put together and tested with no soldering required! Smart Plant also comes with a USB-controlled pump, water flow sensor, plastic tubing for the water, and buttons for control.

Would you say the focus of this project is on education? Can adults benefit from it as well?

Well, everything we build for the maker market is designed for education and learning. Making is education. Making is learning. Building your own projects allows you to innovate around a framework and do wonderful things that we have never thought of.

> "Making is learning. Building your own projects allows you to innovate"

The educational goals for Smart Plant are:

- Learn about the Raspberry Pi and installing software on the Pi
- Connecting up sensors to the Raspberry Pi
- Learning about feedback loops and regulating water to plants
- Understand your indoor environment and what affects it
- Learn about the new technology called the Internet of Things

What sort of role does the Raspberry Pi play here?

The Raspberry Pi is the key component of the system. It is the brain. We could have designed this using a much simpler computer, such as an Arduino, but we wanted to provide the power of the Pi and Linux to allow the Smart Plant to be connected up to the Internet of Things and Alexa. To do that, we needed computing power, and the Raspberry Pi provides all of that in a single inexpensive project.

We've seen mentions of multiple sensors being bundled in with the Smart Plant Pi, what do they bring to this project?

The sensor package provides a whole indoor environmental monitoring system. Temperature, Humidity, Sunlight, UV Index, and one very cool sensor: the Air Quality Sensor. We were amazed at how sensitive this inexpensive sensor was. We could detect all sorts of events in the entire house. One thing to point out is that virtually all of the time the sensor was under 3200 (rated fresh air) and the average was 2727 across the entire period.

Are people able to track the progress of their plant at all?

Absolutely, there are a couple of ways. The first is with the built-in OLED display. It displays status and also flashes alarms when detected. Secondly, you can monitor your plant on the internet by using PubNub and Freeboard.io (a tutorial on how to do that is part of the SmartPlantPi documentation). Here is a live link to our demo plant: **freeboard.io/board/B1kr4y**

How have you found working with the Raspberry Pi in this project? Would you use it for other projects in the future?

The Raspberry Pi is a fabulous platform for designing projects and prototyping products. You can build projects that can do amazing things and make use of the thousands of software packages and sensors available for the Raspberry Pi.

What is our next Raspberry Pi Kickstarter? We are building a board that will allow the maker to experiment with artificial intelligence on the Raspberry Pi with some neural network hardware assists. Can you just imagine what we can build with that? Technology doesn't just have to be the domain of the technologists. We all can be part of this future. Learn. Make. Repeat. ■

John Shovic

has spent over 30 years in hardware engineering, and is a proud author of over 50 papers, including coverage of Arduino and the Raspberry Pi.

Like it?

John can boast a pretty incredible portfolio of work, much of which he goes into detail in over on his site at **switchdoc.com**. Make sure you take a look at the site's tutorial section, which is packed with tips and tricks for beginner and advanced Pi projects alike.

Further reading

The entire process of the SmartPlantPi is fairly complex, and if our brief overview here isn't enough, then make sure you head across to the **switchdoc.com** website where a full rundown is available.

Get hands on with the Pimoroni Blinkt!

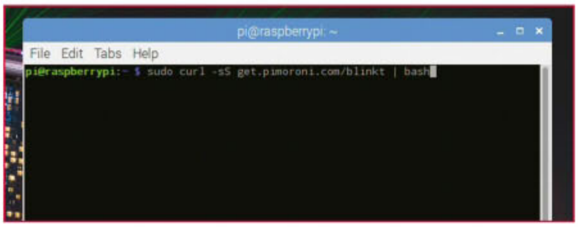
Dan Aldred is a Raspberry Pi Certified Educator and a lead school teacher for CAS. He led the winning team of the Astro Pi secondary school contest and appeared in the DfE's 'inspiring teacher' TV advert. Recently he graduated from Skycademy, launching a Raspberry Pi attached to a high-altitude balloon to over 31,000 metres into the stratosphere.

Learn the basics and create sparkling lights with your Blinkt!

Pimoroni has created an awesome 'super-bright RGB LED indicator' that is ideal for adding visual notifications to your Raspberry Pi without breaking the bank! The Blinkt! offers eight APA102 pixels in the smallest (and cheapest) form factor to plug straight into your Raspberry Pi 3/2/B+/A+/Zero. This tutorial walks you through how to install the Blinkt! and the required Python libraries and modules. Next, create and try out some simple code to control the lights and change the colours and the brightness of the LEDs. Finally, combine these skills together to code an LED colour generator that selects a random LED and a random RGB colour value for the range of eight lights. This creates a psychedelic disco-light setup.

What you'll need

- Pimoroni Blinkt!



```
pi@raspberrypi:~ $ sudo curl -sS get.pimoroni.com/blinkt | bash
```

01 Install the Blinkt!

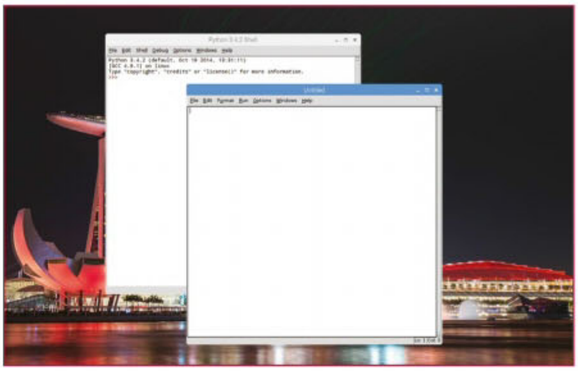
The guys at Pimoroni have made it very easy to install the software for your Blinkt! (which also includes a wide selection of cool examples to show off its features). To get started, ensure that the power is off, attach the Blinkt! to the Raspberry Pi's GPIO pins and push down firmly. One side of it is straight and one is curved. The curved edges align with the curved corners of your Raspberry Pi. This ensures that you attach it the correct way round. Now attach your power supply and boot up your Pi. Open the Terminal and type:

```
curl -sS get.pimoroni.com/blinkt | bash
```

This will install all the required code.

02 Turn on an LED

To turn on an LED, open your Python editor and import the required Python module, line 1. Use the code **set_pixel()**, line 2, to set which pixels are turned on. The first number in the brackets corresponds to each LED. As with the common computing method of numbering, the first LED is referred to by the number zero, the second LED is number one and so on up to the eighth LED, number seven. Copy the code below and run the program to turn the first LED on.



Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Untitled
File Edit Format Run Options Windows Help

```
Python 3.4.2 (default, Oct 19 2014, 13:31:11)
[GCC 4.9.1] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
```

```
from blinkt import set_pixel, show
set_pixel(0,255,0,0) show()
```

WHO IS USING THE INTERNET?

Do you live in a household or work in an office where several people use the internet? Are there numerous devices always connected to the network that suck the bandwidth up and slow down the speed for all other users? Combine the Blinkt with Python and nmap to identify users, track them on the network and create a colourful LED visual for each user as they join and leave the network. Watch the video here: **youtu.be/r_JDw65FTMA**

03 Change the colour of an LED

The next three values in the brackets (x, 255, 210, 150), are used to control the colour of the LED with the standard RGB colour palette. You can control and set the amount of Red, Green and Blue between the values of 0 and 255; the higher the number, the more of that colour is displayed on the LED. Combining these values gives you over 16 million possible combinations and colours. Change the values in your code, save and then run.

```
set_pixel(0,255,255,0) show()
```

04 Adjust the brightness of the LEDs

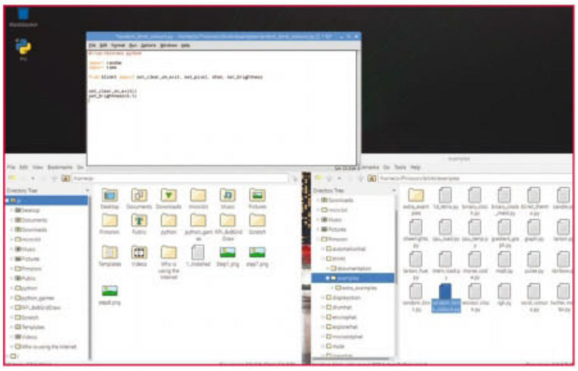
You may find that the LEDs are too bright to look at. It is not advisable to look at them for long periods as they may damage your eyes. Instead, turn down the brightness so that they can be viewed safely. First, import the brightness module, line 1. You can import multiple modules by including the 'module name' on the same line– for example, **from blinkt import set_pixel, show, set_brightness**. Now set the brightness to a suitable level, line 2. The values range between zero and one, where one is full brightness and zero renders all the LEDs off.

```
from blinkt import set_brightness
set_brightness(0.5)
```

05 Turn on multiple lights

Turn multiple LEDs to on using the **set_pixel** code and then set the RGB colour values. Remember that for the LED number, 'zero' is the first LED, as you count from zero upwards. The last LED is number seven. Try turning on two or more LEDs; for example, the last LED set to red and the fourth LED set to blue. Save and run the program.

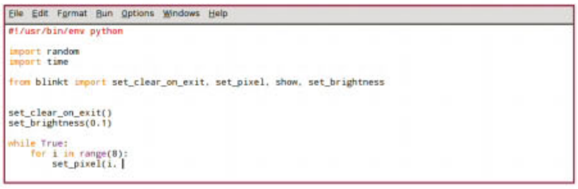
```
set_pixel(7,255,0,0) show()
set_pixel(3,0,0,255) show()
```



06 Create a random blinking light

Combine the previous code steps and create a random set of sparkling disco lights! Begin by selecting a new blank Python file and importing the random module, line 1. This is used to select random values for the colours and the LED position number. Next, import the time module, line 2. This adds control over the frequency of the change (a bigger time value will result in slower blinks). Finally, import the codes to clear the LEDs when the program exits, set a particular pixel to on and reduce the brightness, line 3. Line 4 enables the LEDs to clear on exit and line 5 sets your required brightness value between zero and one.

```
import random
import time
from blinkt import set_clear_on_exit, set_pixel,
show, set_brightness
set_clear_on_exit()
set_brightness(0.1)
```

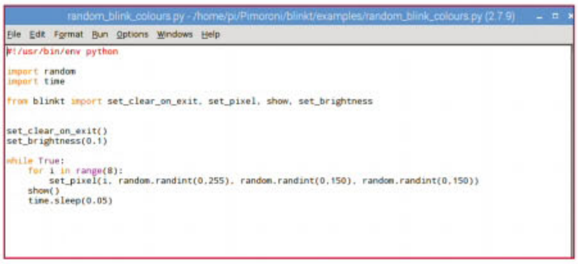


```
#!/usr/bin/env python

import random
import time

from blinkt import set_clear_on_exit, set_pixel, show, set_brightness


set_clear_on_exit()
set_brightness(0.1)

while True:
    for i in range(8):
        set_pixel(i, 
```

07 Create a while loop and set random values

Now create a **while** loop to ensure that the program continually repeats and loops over the code, line 1. On the next line, (which is indented), use a **for** loop to iterate the code over each of the LEDs. This has the effect of applying the next instruction to each of the LEDs, working from the first LED to the last, line 2. Turn the LED on using the **set_pixel** code, line 3; note that this is indented too.

```
while True:
  for i in range(8):
    set_pixel(i,
```



random_blink_colours.py - /home/pi/Pimoroni/blinkt/examples/random_blink_colours.py (2.7.9)
File Edit Format Run Options Windows Help

```
#!/usr/bin/env python

import random
import time

from blinkt import set_clear_on_exit, set_pixel, show, set_brightness


set_clear_on_exit()
set_brightness(0.1)

while True:
    for i in range(8):
        set_pixel(i, random.randint(0,255), random.randint(0,150), random.randint(0,150))
    show()
    time.sleep(0.05)
```

08 Select a random colour

When coding the LED, state the RGB colour values of the LED as covered in step 3; use the code **random.randint(0,255)**. This randomly selects a colour value between 0 and 255. Add two more incidences of the code to select values for the Green and Blue colours. On line 2, set the LEDs to display the colour using the code **show()**; note that this line doesn't have an indent. Finally, add a short time delay so that you can observe the lights before they change state, line 3.

```
  set_pixel(i, random.randint(0,255), random.
randint(0,255), random.randint(0,255))
show()
time.sleep(0.05)
```

09 Run the program

Save your program file and run it by pressing **F5**; call the file a suitable name. Press **Enter** and the program will run, displaying a variation of random colours on each of the LEDs. Experiment with the colour values to create variations that you like and also to speed up or slow down the delay between changes. ■

RED, GREEN AND BLUE COLOURS

The RGB colour model is an additive colour model in which red, green and blue light is combined in various amounts to create a wide range of colours, over 16 million different variations. The lowest value for a colour is zero, which usually denotes black, as black is not a colour but an absence of colour! The top value is 255; for example, red 255 means the maximum amount of red is being added to the shade.

Make a Raspberry Pi VPN Access Point

Use your Pi as a Wireless AP that's connected to your VPN 24/7

Nate Drake
(@natewriter)
is a freelance technology journalist specialising in cybersecurity and doomsday devices.

Anyone who has been abroad will know it's a constant pain to have to deal with content that has been switched to the local language, not to mention certain states that censor some internet content.

Fortunately there's a way to evade these restrictions using the Raspberry Pi. In a few simple steps you can configure your Pi to generate its own wireless AP (Access Point) and keep it permanently connected to a VPN (Virtual Private Network) service.

All you need to do when travelling is bring your Pi and connect it to a working router and you'll have your own private wireless network and connection.

VPNs were originally designed to allow office workers to connect to their corporate intranet while away, over an encrypted connection. These days they're more commonly used both to protect your connection and make it seem as if your computer is located in another country.

In order to proceed with this project, you will require an active VPN subscription and the client configuration (.conf) file to automatically connect.

01 Choose your VPN

Best VPN Services to Buy with Bitcoin

In order to carry out this project, you need an account with a VPN provider. Find a provider that supports the OpenVPN protocol, as this connection is generally considered to be the most secure. Free providers won't require any billing information but they are not as fast or reliable as paid services. If you choose a paid provider, try to find one that accepts anonymous payment methods such as Bitcoin. The website **www.weusecoins.com** has a list of these.

02 Download VPN configuration files

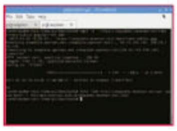
Attach the Pi to your router. Open Terminal or connect via SSH. If your VPN provider supports the secure OpenVPN standard, then they will have provided a configuration file with the extension .conf or .ovpn. For this tutorial a VPN configuration file from free provider VPNBook was used. Download the file to your Pi either by clicking the link or using wget in Terminal, for instance:

What you'll need

- Raspberry Pi with Wi-Fi support
- Ethernet cable
- An active VPN subscription that supports OpenVPN
- Secondary device such as a laptop to test your AP works

```
wget http://www.vpnbook.com/free-openvpn-account/VPNBook.com-OpenVPN-Euro1.zip
```

03 Install OpenVPN



The Raspbian repositories contain the OpenVPN software but not the most current version. Use the command su to switch to the root user then run these commands:

```
wget -O - https://swupdate.openvpn.net/repos/repo-public.gpg|apt-key add -
```

```
echo "deb http://swupdate.openvpn.net/apt jessie main" > /etc/apt/sources.list.d/swupdate.openvpn.net.list
```

```
apt-get update
```

```
apt-get install openvpn
```

Run **openvpn –version** to double-check you have the most up-to-date version of the software. At the time of writing this is 2.3.14.

04 Configure and run OpenVPN

Use the mv command to move the configuration file into the openvpn folder /etc/openvpn, amending the extension if necessary, for instance:

```
sudo mv vpngate_vpn151111650.opengw.net_udp_1344.ovpn /etc/openvpn/vpn1.conf
```

Next, start the OpenVPN service with the command sudo service openvpn start. Start OpenVPN using your .conf file with the openvpn --config command, for instance:

```
sudo openvpn -config /etc/openvpn/vpn1.conf
```

Next, run

```
sudo service openvpn start
```

05 Test OpenVPN connection

Once the OpenVPN service is running, open a new tab in your Terminal or start a new SSH service and run the command ifconfig to list your network interfaces. Usually the VPN connection will appear as tun0.

You can check your apparent location with the command:

```
curl --interface tun0 freegeoip.net/json/
```

Next, make sure the OpenVPN service starts each time you log in. Then run

```
sudo nano /etc/default/openvpn
```

and remove the # at the start of the line reading "#AUTOSTART="all".

06 Install prerequisites

Now we'll install the necessary software to set up a Wireless AP.

Do this by running

```
sudo apt-get install dnsmasq hostapd
```

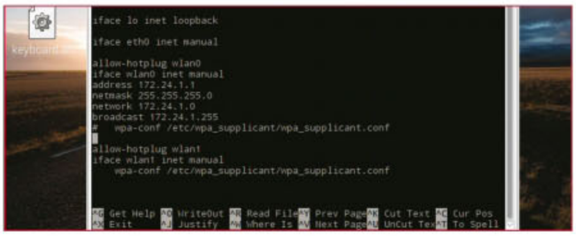
Next, run

```
sudo nano /etc/dhcpcd.conf
```

Add these lines to the very bottom of the file:

```
interface wlan0
static ip_address=172.24.1.1/24
```

Press Ctrl+X, Y then Return to save and exit.



```
iface lo inet loopback

iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet manual
address 172.24.1.1
netmask 255.255.255.0
network 172.24.1.0
broadcast 172.24.1.255
#   wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

07 Set static IP

Open your network interfaces configuration with

```
sudo nano /etc/network/interfaces///ENDCODE
```

Change the line "iface wlan0 inet static" to "iface wlan0 inet manual". Press Return to start a new line, and then paste:

```
address 172.24.1.1
netmask 255.255.255.0
network 172.24.1.0
broadcast 172.24.1.255
```

Place a '#' at the start of the line beginning "wpa-conf". Save and exit in the same way as before.

Restart the dhcpcd service with

```
sudo service dhcpcd restart
```

08 Configure AP

Run

```
sudo nano /etc/hostapd/hostapd.
conf'
```

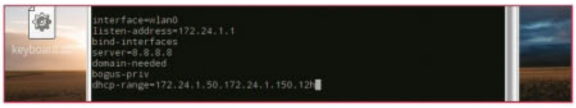
and paste the following:

```
interface=wlan0
driver=nl80211
ssid=piVPN
hw_mode=g
channel=1
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_passphrase=raspberry231
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Feel free to change the name of the network from 'PiVPN' to one that is meaningful to you. Similarly change the password 'raspberry231' to something more secure.

```
Next run nano /etc/default/hostapd
```

Find the line starting #DAEMON_CONF="" and change to DAEMON_CONF="/etc/hostapd/hostapd.conf. Note the '#' at the start of the line must be removed.



```
interface=wlan0
listen-address=172.24.1.1
bind-interfaces
server=8.8.8.8
domain-needed
bogus-priv
dhcp-range=172.24.1.50,172.24.1.150,12h
```

09 Configure dnsmasq

Move the old dnsmasq configuration file with

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

then create a new one by running

```
sudo nano /etc/dnsmasq.conf
```

Paste in the following text:

```
interface=wlan0
listen-address=172.24.1.1
bind-interfaces
server=8.8.8.8
domain-needed
bogus-priv
dhcp-range=172.24.1.50,172.24.1.150,12h
```

Note we are using Google's DNS server (8.8.8.8) for now; change this if you wish, then save and exit.

Run

```
sudo nano /etc/sysctl.conf
```

Automatic Login

If you need a username and password for your VPN, you can save these so OpenVPN will connect automatically.

First run

```
sudo nano auth.txt
```

On the first line put the username and on the second put your password. Save and exit. Next edit your OpenVPN config file e.g.

```
sudo nano /etc/
openvpn/vpn2.conf'
```

Scroll down to the line with the text "auth-user-pass". Leave a space and enter the path auth.txt, for example:

```
auth-user-pass /
home/pi/auth.txt
```

Save and exit once again.

Find the line starting "net.ipv4.ip_forward=1" and remove the '#' at the start. Now reboot the Pi.

10 Set up IPV4 Forwarding

For the next step, you need to run each of these commands individually:

```
sudo iptables -t nat -A
POSTROUTING -o eth0 -j MASQUERADE
```

```
sudo iptables -A FORWARD -i eth0 -o wlan0 -m state
--state RELATED,ESTABLISHED -j ACCEPT
```

```
sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

Next, run

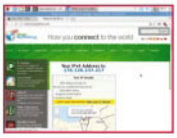
```
sudo nano /etc/rc.local
```

Paste the following right above the line reading "exit 0":

```
iptables-restore < /etc/iptables.ipv4.nat
/usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

11 Test Access Point

Run the commands



```
sudo update-rc.d hostapd enable
```

and

```
sudo update-rc.d dnsmasq enable
```

Reboot the Pi. You'll need a second device at this stage to see if you can access the Wireless AP. Search for it in your network menu and enter the password you created earlier on. If you can't remember this, run

```
sudo nano /etc/hostapd/hostapd.conf
```

on the Pi to view it again. Once connected visit **www.whatismyipaddress.com** to check you're behind the VPN.

12 Fix DNS Leaks

Hello 176.126.237.217

Certain VPN providers use their own DNS servers. Other VPN Providers are less cautious. Visit **https://www.dnsleaktest.com/** and click Extended Test to check you're safe. If any of the DNS servers match your regular ISP, your connection is not fully secure.

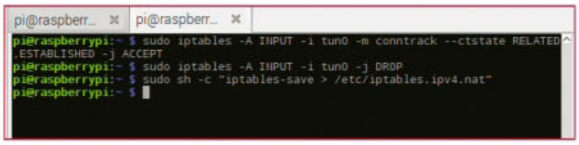
To resolve this, edit your VPN configuration file e.g.

```
sudo nano /etc/openvpn/vpn2.conf
```

and add these lines immediately above "<ca>":

```
script-security 2
up /etc/openvpn/update-resolv-conf
down /etc/openvpn/update-resolv-conf
```

Save and exit, then restart your Pi. Check the DNS leak website once again. If this fails to resolve the issue, try using another VPN provider.



```
pi@raspberrypi:~ $ sudo iptables -A INPUT -i tun0 -m conntrack --ctstate RELATED
,ESTABLISHED -j ACCEPT
pi@raspberrypi:~ $ sudo iptables -A INPUT -i tun0 -j DROP
pi@raspberrypi:~ $ sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
pi@raspberrypi:~ $ 
```

13 Block unsolicited connections

As your Pi is sitting between your computer and the internet, it can potentially be accessed by other devices.

Prevent unsolicited incoming connections from other devices with the following commands:

```
sudo iptables -A INPUT -i tun0 -m conntrack
--ctstate RELATED,ESTABLISHED -j ACCEPT
```

```
sudo iptables -A INPUT -i tun0 -j DROP
```

Make sure to save your changes so that they'll apply on reboot:

```
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

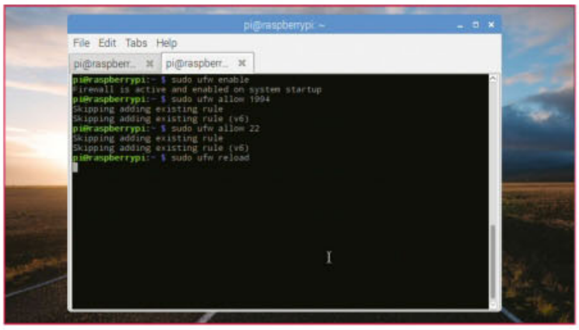
14 Route all traffic through OpenVPN

When you boot the Pi initially, certain applications may try to connect directly to the internet, which can undermine your anonymity.

To channel all network traffic through the VPN, you need to edit your configuration file in /etc/openvpn, for instance by running

```
sudo nano /etc/openvpn vpn2.conf
```

Make sure the line "redirect-gateway" reads "redirect gateway def1". DNS queries will also be routed through the VPN also so make sure your provider supports this.



```
pi@raspberrypi:~ $ sudo ufw enable
Firewall is active and enabled on system startup
pi@raspberrypi:~ $ sudo ufw allow 1994
Skipping adding existing rule
Skipping adding existing rule (v6)
pi@raspberrypi:~ $ sudo ufw allow 22
Skipping adding existing rule
Skipping adding existing rule (v6)
pi@raspberrypi:~ $ sudo ufw reload
```

15 Set up Firewall

Although you may have previously configured iptables to prevent unsolicited incoming connections, to be on the safe side, consider installing ufw (Uncomplicated Firewall) with

```
sudo apt-get install ufw
```

Run

```
sudo ufw enable
```

to fire it up, then open the default OpenVPN port 1194 with

```
sudo ufw allow 1194
```

You may also want to enable Port 22 to allow connecting via SSH. Remember that this doesn't change which ports are open and closed on the router. ■

The Electronic Frontier Foundation is the leading nonprofit organization defending civil liberties in the digital world. Founded in 1990, EFF champions user privacy, free expression, and innovation through impact litigation, policy analysis, grassroots activism, and technology development. We work to ensure that rights and freedoms are enhanced and protected as our use of technology grows.

EFF.ORG

ELECTRONIC FRONTIER FOUNDATION

Protecting Rights and Promoting Freedom on the Electronic Frontier

Anaconda for the Pi

Python is the language of choice on the Raspberry Pi, and there are many packages available within the Raspbian repository. Berryconda helps you move out beyond these modules

Joey Bernard
is a true renaissance man, splitting his time between building furniture, helping researchers with scientific computing problems and writing Android apps

Why Python?

It's the official language of the Raspberry Pi. Read the docs at **python.org/doc**

Python has always been the language of choice for projects on the Raspberry Pi. There are several reasons for this, but we won't dive into the history here. As a consequence of this decision, there are many Python modules available as packages within the Raspbian package repository. You can trust that these have been compiled to run on the ARM processor and that they have been tested and are therefore safe to use. But not all Python modules are available this way. For the missing modules, the intention is that you could use pip to install them yourself on your Raspberry Pi. While this works fine for some modules, you will eventually start running into those modules that require debugging to figure out why they aren't working correctly on the ARM processor. The amount of work involved in debugging these types of issues really should be distributed across many people, and luckily it is. For Python coders, a very good distribution of Python modules is available as the Anaconda project. There is also a community port of the Anaconda project, called Berryconda, which has been ported to the Raspberry Pi. The main website for the project is located at github.com/jjhelmus/berryconda. This article will cover the steps to getting it installed and ready to use on your Raspberry Pi so that we can go on and look at some of the functionality that becomes available to you for Python coding in future articles.

The very first step is to download and install the core of the Berryconda system. On the download page, there are installation files for both armv6l (Raspberry Pi 1 or Zero) and armv7l (Raspberry Pi 2 or 3). There are also different versions of the installer for Python 2.X or Python 3.X. This covers all of the options that you might need for your particular project. The installer is actually a shell script, so all you need to do is to make the script executable, and then run it. For example, we can install the Python 3.x version on a Raspberry Pi 1 with the following commands:

```
wget https://github.com/jjhelmus/
berryconda/releases/download/v1.0.0/
Berryconda3-1.0.0-Linux-armv6l.sh
chmod +x Berryconda3-1.0.0-Linux-
armv6l.sh
./Berryconda3-1.0.0-Linux-armv6l.sh
```

By default, this will install the base of an Anaconda environment into the directory **berryconda3** in your home directory. You can change this installation directory during installation in case you want to put it somewhere else. The installer also asks you whether you want to add the path to the binaries to your PATH environment variable. This is generally a good idea. Just remember to exit from the current shell and log back in so that the new path is picked up. Either that, or manually source the initialisation file to get it set.

> "The main utility in Anaconda is the conda packaging system"

The main utility within Anaconda is the conda packaging system. Conda provides a very fully featured package management system to handle Python modules and all of the various dependencies that may be required. There is a very good set of documentation available at the website **conda.io/docs**, covering all of the options and functionality available. The very first thing you will want to do is to keep your current system up to date. You can update individual packages with the command:

```
conda update package-name
```

This command will go out to the internet and figure out what new versions of the given package exist, and if it finds one, it will ask you whether you really want to perform the update. If you want to just see what will be done, you can use the command line option **--dry-run** to get a display of what commands would be run. If you just want to keep the entire system updated, you can use the following command to handle updating everything:

```
conda update --all
```

To install new packages, you need to first find out what has already been packaged and is available. Because this is a community effort, the selection will vary over time. So you should check before trying to install some packages. You can do that search with the command:

```
conda search some_text
```

This will do a regular expression search of package titles, looking for the given text, and return a list of any available. Sometimes, however, you may get a rather large list returned. If this happens, you can do a more refined search using the usual regex options used in many other UNIX utilities. If you already know the package name, you could use the **-f** option to force the search to only return exact matches to the text you give it. If the package in question is already installed on your Raspberry Pi, it will have an asterisk beside it for the version installed. When you are ready to install the package you were searching for, you can do so with the following command:

```
conda install ipython
```

This will, by default, install the latest version of the ipython package within the Berryconda environment. It will also install any missing dependencies, as well as updating any out-of-date dependencies.

While the online documentation is great, there are also help pages available within conda. If you want general help, you can get it with the **help** command. For help with some specific conda commands, you can use something like the following command:

```
conda install --help
```

This is handy for all of the details that nobody can seem to remember. Over time, you may find that not all of the installed packages are needed any more. This could be an issue on a small machine like the Raspberry Pi. The first step is to clean up the Berryconda environment. You can remove unused versions of packages, cached installation tarballs of the packages in the environment, and index caches. Assuming that you simply want as lean a system as possible, you can clean up the entire environment with the following command:

```
conda clean --all
```

If there are installed packages that are no longer needed, you can remove them with the uninstall command. For example, the following command would remove the scipy Python module:

```
conda uninstall scipy
```

If you have a drastic need to restart, you could remove everything from the environment by using the **--all** command option.

One of the great strengths of Python is the large set of third-party modules available for extending functionality. Unfortunately, this is also one of its weaknesses, leading to a large amount building up over time. The Python module python-env was written to try to deal with this issue. The conda packaging system also understands creating isolated Python environments. This way, you can have the best of both worlds: easier Python module management combined with easier Python environment management. You can get a list of all of the environments conda knows about with the following command:

```
conda env list
```

If you have just installed Berryconda, the only environment listed is the root environment. Now, let's say that you needed to start a new code base, developing the software for a big new project. You could create a new empty environment with the following command:

```
conda create --name deathstar
```

This command creates a new subdirectory within the **envs** subdirectory inside the Berryconda environment. If you rerun the environment list command, you will now see both the root environment and the deathstar environment, with the latter tagged by an asterisk as the currently active one. You can now install only the modules you require for this specific software project by activating it and running the usual conda commands. For example, the following commands would install ipython, along with all of its dependencies, within the deathstar environment.

```
source activate deathstar
conda install ipython
```

You can leave a given environment by running the deactivate script to reset all of the environment variables. If you have already put together a basic environment that you want to use as a starting point for a new environment, you can clone it as your boilerplate. The clone option to the create command comes in handy for such a task.

```
conda create --name deathstar2
--clone deathstar
```

If the reason for your software project goes away, you can always clean up old environments with the remove command in conda.

```
conda --remove deathstar --all
```

The command option **--all** ensures that all portions of the environment are deleted during the removal process.

Now, you should have all of the information you need to be able to work in isolated environments for your Raspberry Pi projects. This is especially useful in the development stages of a new project. You can get only the modules you need for a given project.

Raw python-env

While the conda system is a very powerful tool in simplifying the management of environments and Python modules, it isn't the only one available. As mentioned in the main article, there is the python-env module available to do the same basic task. You can install it with the command:

```
sudo apt-get install virtualenv
```

It will also install a number of dependencies. The main command is called virtualenv. With it, you can create new environments, manage them and even delete unneeded ones. To create a new environment, you can use the following command:

```
virtualenv deathstar
```

This will create a new subdirectory, named deathstar, within the current directory. It then makes a copy of the core of a Python environment, including Python itself and setup tools like pip. There is also a set of management scripts available within the newly created environment. In a similar fashion to conda environments, you can activate the 'deathstar' environment with the following command:

```
source ./deathstar/bin/activate
```

Once you activate the new environment, any Python modules you install will be installed within this environment's directory structure. For example, if you install scipy with the command:

```
pip install scipy
```

...all of the files associated with the scipy module will be installed within the lib subdirectory of the environment. Depending on what versions of pip and setuptools are initially available when you created your new environment, you may need to update them before installing any new modules. If you run into any issues, start with the following two commands before diving too deeply into debugging:

```
pip install -U pip
pip install -U setuptools
```

This gives you another option if you don't want to deal with the full conda system.

BUILD A BETTER WEB

www.webdesignermag.co.uk

web designer
FREE 115 MINUTES OF JS VIDEOS
CSS LAYOUTS
<CODE> <SMALLER> <SMARTER> HTML
HOW TO UNLEASH THE POWER OF WEB COMPONENTS
SUPERFAST WEBSITES
CODE WITH CSS SHAPES
UNIVERSAL JAVASCRIPT APPS
GET INSTANT ANIMATION
CUSTOM WP MENUS
FREE 73 MINUTES OF CSS VIDEOS
FREE 84 MINUTES OF TYPE VIDEOS
25 VITAL APIs
TOOLS

EVERY ISSUE • FREE RESOURCE DOWNLOADS

web designer

Available from all good newsagents and supermarkets

ON SALE NOW

Industry interviews | Expert tutorials & opinion | Contemporary features | Behind the build

DESIGN INSPIRATION

PRACTICAL TIPS

BEHIND THE SCENES

STEP-BY-STEP ADVICE

INDUSTRY OPINION

BUY YOUR ISSUE TODAY

Print edition available at www.imagineshop.co.uk

Digital edition available at www.greatdigitalmags.com

Available on the following platforms

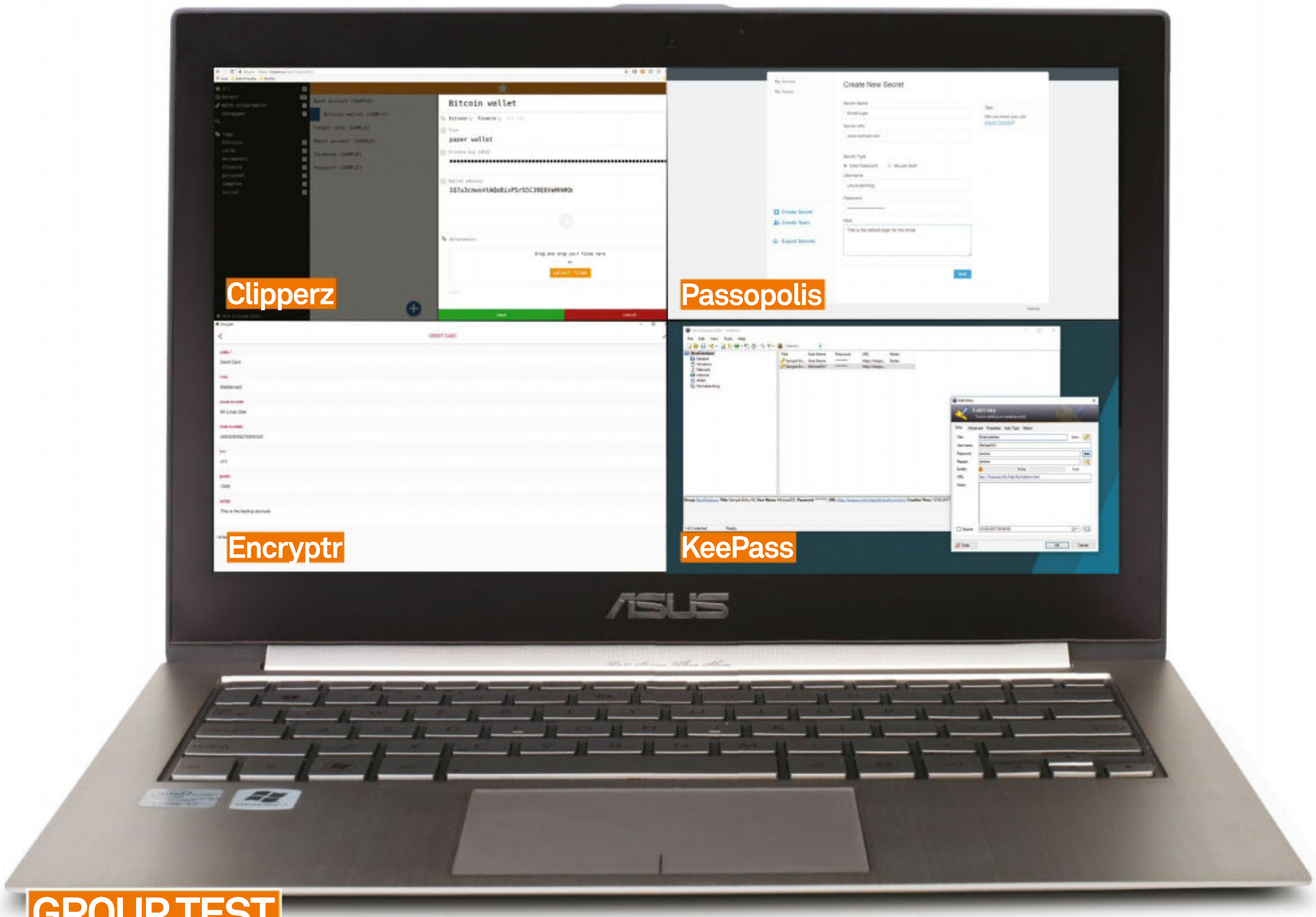
Print | Tablet | Phone | Desktop

facebook.com/webdesignermag

twitter.com/webdesignermag

Reviews

83 **Group test** | 88 **Synology RT2600ac** | 90 **AryaLinux 2017** | 92 **Free software**



GROUP TEST

Password managers

Keeping the details of your accounts safe is of more importance than ever before, but which open-source password manager is worth investing your time in?

Clipperz

Security is key when it comes to Clipperz, which is why it claims to know nothing about their users or the data they choose to encrypt. Through passphrases and vault options, Clipperz looks to offer everything a user could need to keep their virtual presence safe. Clipperz.com

Passopolis

While most password managers are tailored to individuals, Passopolis uses a friend-based system where users are able to invite certain people to access their account. It's a unique feature, but does the program has a whole do enough to stand out from some fierce competition? Passopolis.com

Encryptr

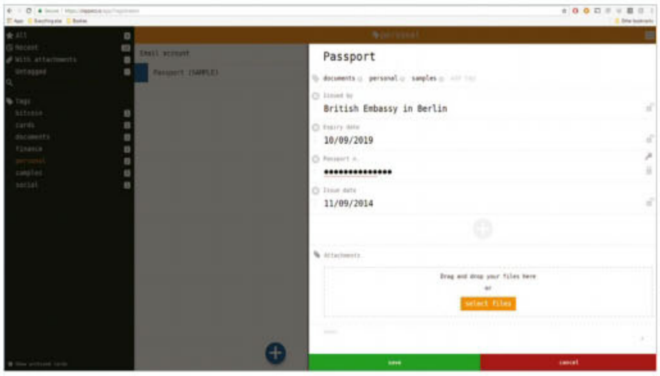
Considering Encryptr comes from the team who produce the amazing Spideroak secure cloud server, Encryptr has a pretty impressive history. It prides itself on its simplicity to attract new users, while not skimping on the staple elements that every password manager needs. Spideroak.com

KeePass

Consider KeePass the old man of the group, and a program that many seasoned Linux users will recognise. It's undergone some big changes as of late, with improvements made to both the UI and expanding on what used to be a fairly limited set of features. Keepass.info

Clipperz

Anonymity is key when it comes to using Clipperz's vault system



Passport

Issued by
British Embassy in Berlin

Expiry date
10/09/2019

Passport #
••••••••••••••

Issue date
11/09/2014

■ Different tags can be applied to each entry you make within Clipperz

UI and design

Thanks to its minimal UI system, navigating through Clipperz is a breeze. Individual vault entries are categorised through keywords, which are then accessible through the side menu; although some didn't save correctly at times. Each password entry can be cloned and archived when needed, a nice touch if you've got several closely linked accounts.

Adding and customising entries

Large amounts of details can be attached to each entry, especially when it comes to debit and credit card information. As you create passwords, Clipperz will indicate any password that it feels needs to be updated, even providing examples for what makes a good password. However, the constant reminders of how rubbish our password were soon became tiresome!

Security

Instead of storing readable data on their server, they use a host-proof web app that instead encrypts and decrypts the data inside Clipperz. This is great news for users, as it makes it near impossible for your entries to be compromised from their side. There's also 128-bit encryption board, which automatically adds two-step authentication at any point where Clipperz deems it necessary to use.

Extra features

One of Clipperz's best features is its one-time password system, which allows for secure access to your files from an insecure device. It works a treat, and especially useful when you need instant access when out and about. There's also a mobile version for users to explore, but it lacks many of the core features that the desktop version offers.

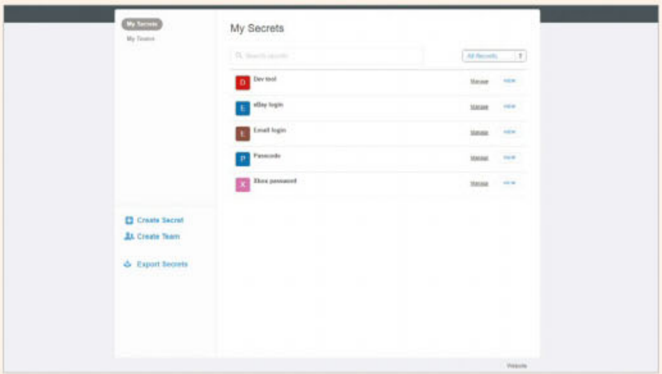
Overall

Clipperz does everything it sets out to do relatively well, and we truly believe your details will be safe. Certain features are a little rough around the edges, but expect that to be sorted in future updates.

8

Passopolis

Is sharing the key to success for Passopolis?



■ Changes to your passwords can be made at anytime

UI and design

While some will be attracted to Passopolis's minimal design, it's a little too barebones for our liking. Most options are available through its on-board account system, but some are hidden away and pretty difficult to find unless you're familiar with the software. However, a handy search tool is available for quickly sifting through your entries.

Adding and customising entries

Known as 'secrets' in Passopolis, adding new entries is a one-button process. Details can be added as you see fit, but it does lack the customisability of what Clipperz offers. However, we did like the team-based entries that can be created, allowing you to share access to specific entries with pre-targeted family and friends when needed.

Security

Although there's nothing truly unique about the security protocols used within Passopolis, it does include everything that you could need. A password generator is on hand for adding custom passwords, while users also have access to two-factor authentication when needed. Exporting files does require a decryption run-through first, which can take a while to complete, however.

Extra features

Again, there's not a lot to get excited about here. There's an import system on hand, however. If you're moving from your previous manager across to Passopolis, you can have your entire vault of entries imported with relative ease. Be warned, this isn't a quick process, and a full import can take anywhere from a few minutes to a couple of hours.

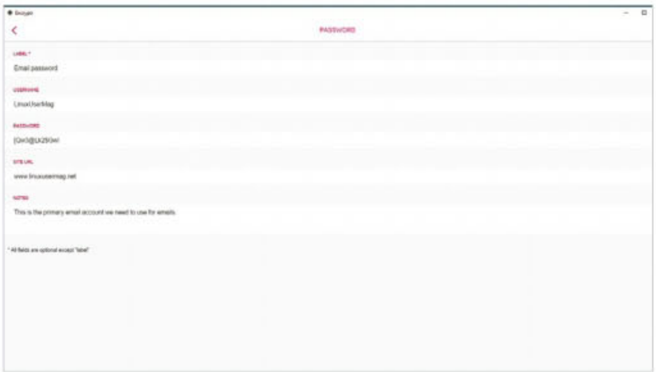
Overall

Sticking to the basics is all well and good, but Passopolis tends to feel a little too minimal at times. A lack of customisation on entries is an annoyance, but all base options are present.

6

Encryptr

Simplicity is key when it comes to Encryptr keeping your files safe



LABEL *
Email password

USERNAME
LinuxUserMag

PASSWORD
[illegible]

SITE URL
www.linuxusermag.net

NOTES
This is the primary email account we need to use for emails

* All fields are optional except "label"

■ Add as much detail as you see fit to any entry created within Encryptr

UI and design

Encryptr prides itself on being lightweight, so the first thing you'll notice is just how fast it's to navigate through. This does mean there's certain cutbacks, and there's not a whole lot to look at once you've booted it up. However, all key options are present and easily accessible through their dual-menu system that's available through the homescreen.

Adding and customising entries

Users have the option to tailor their entry into one of three specific types, depending on if you want to add a payment card, password or general account details. Each menu is crafted with different options and menus to fill in, which makes each of these highly customisable. Best of all, these entries can be merged if you prefer having one steady stream of entries.

Security

Encryptr's ties with security specialists, Spideroak, mean security plays a big role here. A password generator system can be created to make each entry you add truly unique, all of which are backed up to their own Zero Knowledge cloud. One caveat, and something most of its competitors offer, is the lack of two-factor authentication as a staple part of the security process.

Extra features

If you own multiple devices sporting more than one OS, you're in luck! Encryptr works across nearly every OS you can think off, and any details stored within are accessible through each account. For peace of mind, an email will be sent to you every time a new device looks to gain access, which will put some people's minds at rest. We did notice, though, that the warning email is usually sent to your spam inbox, which is something to keep in mind.

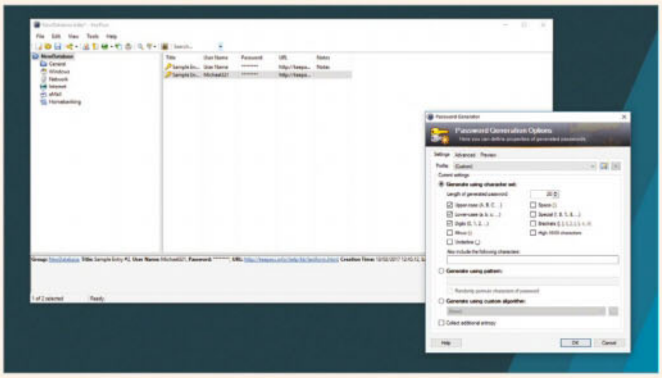
Overall

There's a lot to like in Encryptr, and in terms of both usability and accessibility, it's second-to-none. Some other password managers could learn a lot from what it offers.

9

KeePass

Can KeePass roll back the years and prove it's still a worthwhile manager?



■ KeePass offers a steep learning curve to its users, as there's a lot here to explore

UI and design

Compared to the competition, KeePass is far more convoluted in its design. While the UI is still accessible, there's a vast array of menus and locating specific ones is especially tiresome at first, especially once you start exploring the program's sub-menu system. Although it does take a while to get used to, there's everything you could possibly want here.

Adding and customising entries

KeePass's crowning glory is that it offers the best entry creation system out of the programs featured here. Each entry can be prioritised for the level of protection it requires, as well as implementing a time to remove the entry after a certain date. Basic entries can use the auto-type feature that automates much of the creation process for you.

Security

All the standard security protocols are implemented well here, and the scope of security you apply to each entry is entirely user-dependent. Also included is a master key option, which can be used to secure your account when being accessed from a new machine. We highly recommend using this feature if you've got access to a smartphone to link your account to.

Extra features

File corruption is a possibility within password managers, but KeePass is one of the few that offers a repair system. If this happens and your files are corrupted, KeePass will look to restore the missing entries whenever possible, but be warned, this process can be hit and miss with its results. Multi-user support is also listed, but it's a surprisingly hard system to setup that many will bypass.

Overall

If users are able to look past KeePass's steep learning curve, this is one powerful password manager vault. Every option you could possibly want is included, and some noticeable extras are there as well.

8

In brief: compare and contrast our verdicts

	Clipperz		Passopolis		Encryptr		KeePass	
UI and design	Vault entries are categorised by keywords, making navigation super easy	9	A minimal approach is one thing, but this is a little too barebones for our liking	5	An impressive dual-menu system is used to help keep all the different key tools separated	9	Fairly complicated to navigate around, with a steep learning curve for new users	6
Adding and customising entries	Entries can be as detailed, or as basic, as you deem necessary for your account	7	The creation process is easy, but lacks some core options we'd expect to see	6	Every entry can be filed into one of three types, with each having their own set of options	9	Each entry can be specifically tailored to your exact needs and choice of security	9
Security	No readable data is stored on their server, so there's little chance of being compromised	8	Provides everything you need to keep your account details safe and secure	7	No two-factor support here is an annoyance, but all the other options are here	7	Different layers of security can be applied to every entry that you create	8
Extra features	One-time passwords are ideal for accessing your files on an insecure device	8	The import system is handy, but incredibly slow when it comes to it working	6	Multi OS support means that you can access your account wherever you may be	9	While a repair system is on hand for any corrupted files, it's a little hit and miss	7
Overall	**Clipperz offers a unique approach to what a password manager should be**	**8**	**Passopolis, in its current state, fails to compete with the stronger competition available**	**6**	**Encryptr has brought everything we love into one handy program that every user should have**	**9**	**KeePass is credible, but it's not recommended for those not used to this sort of software**	**8**

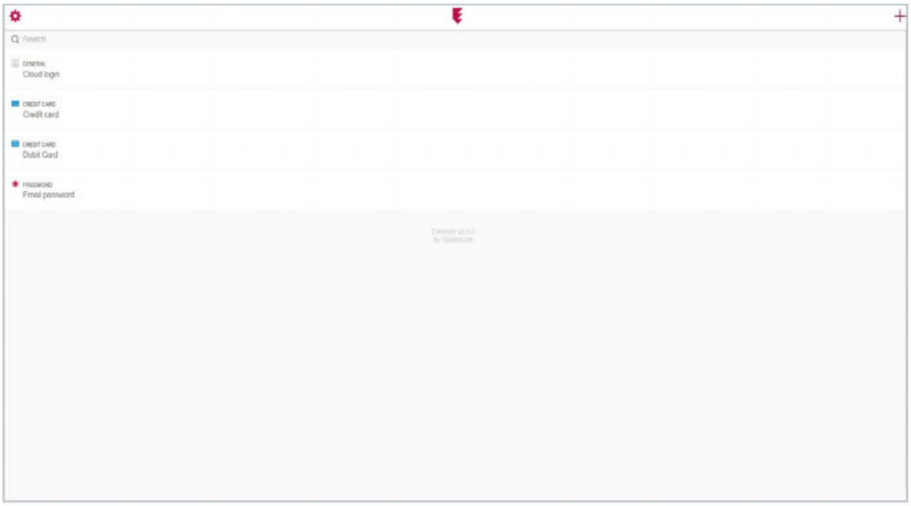
AND THE WINNER IS…

Encryptr

The core technology behind password managers has dramatically changed over the past 24 months, and perhaps the most pleasing element about undertaking this group test is that some of the best Linux offerings have also moved with the times. Each of the four open-source Linux password managers we tested here all have their strong points, but it was Encryptr that really impressed us.

Simplicity plays a big role in Encryptr's success, and everything from the initial set up process to adding your first entry can be done in a matter of minutes. Although we should mention that advanced users can still find what they need here, with many additional options for tailoring the program to your desktop also available.

Each entry you make within the program will fit into one of three categories, with each sporting their own set of options to help fill in all the necessary details. It's something that you don't necessarily find in other management systems, but it works a treat here. Of course, it's also simple to merge all your entries together, if and when you need to.



Entries can be merged into one, easily accessible list when needed

It's no slouch in the security department either, with Encryptr's password generator on hand to make sure your details are kept safe. The program's links with Spideroak also plays a massive role, helping you and your account stay completely anonymous when you use it. One omission is two-factory authentication, but it's something we expect to see later.

While we highly recommend trying out both KeePass and Clipperz as a viable alternative password managers, it's Encryptr's scope of features and usability that really make it stand out from the crowd. There really is no better way to keep your password and account details safe.

Oliver Hill

HARDWARE

Synology RT2600ac Wireless Router

Synology's latest router is an expensive piece of equipment – can it justify the price tag?

Price
£243.03

Website
ebuyer.com

Specs
802.11a/b/g/n/ac wireless
Simultaneous 2.4GHz/5GHz operation
4 Gigabit Ethernet ports
1 USB 3.0 port
1 USB 2.0 port
SD card slot

There's something pleasantly old-school about the visual appearance of Synology's latest router. It's a large piece of kit, and draws attention to itself with four very prominent aerials – suffice to say it's not aiming to be as pretty and living room-friendly as the average ISP-issued wireless router. Synology has clearly focused on delivering an outstanding piece of networking kit, and in that respect it has delivered.

Everything from initial set-up to ongoing management is handled through Synology Router Management, a web-based GUI that mimics the look and feel of a desktop operating system. With a minimum of fuss, we had the device working in harmony with a cable modem and connected a full household of devices in less than half an hour. The GUI provides instant familiarity, and helps you sort through the settings available – of which there are an enormous number.

As an example of the level of detail allowed in customisation, parental controls include device-specific time limits and website blocking, while the LEDs on the front of the router can be configured to avoid distracting blinking on a day-by-day, hour-by-hour basis. Other useful functions include a built-in

Above The RT2600ac has a relatively wide body, over twice the size of some of the smaller routers on the market

Above The four angular aerials draw attention to themselves, meaning that you'll probably want the RT2600ac out of sight

> Whether you want simple and secure networking or a full-fledged NAS, the Synology RT2600ac will easily meet your needs

firewall, DoS protection, and auto-blocking. The level of control on offer is simply remarkable.

The device is also capable of acting as a home file server, thanks to its support for external storage via USB or SD card. With some simple configuration, this also allows the device to function as a UPnP media server for streaming to your smart TV, game consoles and mobile devices – a feature we successfully enjoyed with a binge-watch marathon! The Cloud Station Server function is also useful, allowing you to synchronise select files between a number of connected devices.

As the device operates both 2.4GHz and 5GHz simultaneously, connection is simple and the router will automatically determine which connection is most appropriate for each specific device connected. In our usage tests, speed was faultless with ten or more devices connected, and the wireless range was easily sufficient to cover standard household usage and more, with portable devices connecting some way out into the garden.

Users of wired networks have access to four gigabit Ethernet ports, though the first of these can be used as a second WAN port. This can be configured as a failsafe connection for instances in which the primary WAN connection is unavailable, or as a second source of bandwidth – in this case, the load is balanced in a 60/40 split with the primary WAN port handling the larger share of the work.

There are very few drawbacks to the Synology RT2600ac. The biggest is the price – you'll need to be very serious about home networking in order to spend more than £200 on a router, no matter how good it is. If you've already invested in lots of cabling and you're just looking to replace your existing router, you may also find that the standard inclusion of four Ethernet ports is a tad miserly, especially considering the size of the unit. If you're using a second WAN connection, this falls to just three ports, meaning that you might end up needing an additional network switch.

These drawbacks pale in comparison to the benefits of the device, though. For your investment, you get a wireless router that offers faultless range and speed, which can quickly and comfortably be configured to your exact requirements. Whether you want simple and secure networking or a full-fledged NAS with all the bells and whistles imaginable, the Synology RT2600ac will easily meet your needs. Just don't expect it to look pretty while it does so.

■ ***Nicholas Thorpe***

Pros

Great performance on both 2.4GHz and 5GHz bands, and a user interface that sets it well apart from competing routers

Cons

It's not cheap, and heavy users of wired networking might have hoped for more Ethernet ports. It's also pretty bulky

Summary

If you're looking to create a home network, you can't go wrong with the Synology RT2600ac – the speed and wireless range are good, and it's rare to see a router that offers so much flexibility while maintaining a user-friendly interface. It's a costly unit, but the feature set ensures that it's worth every penny.

9

DISTRO

AryaLinux 2017

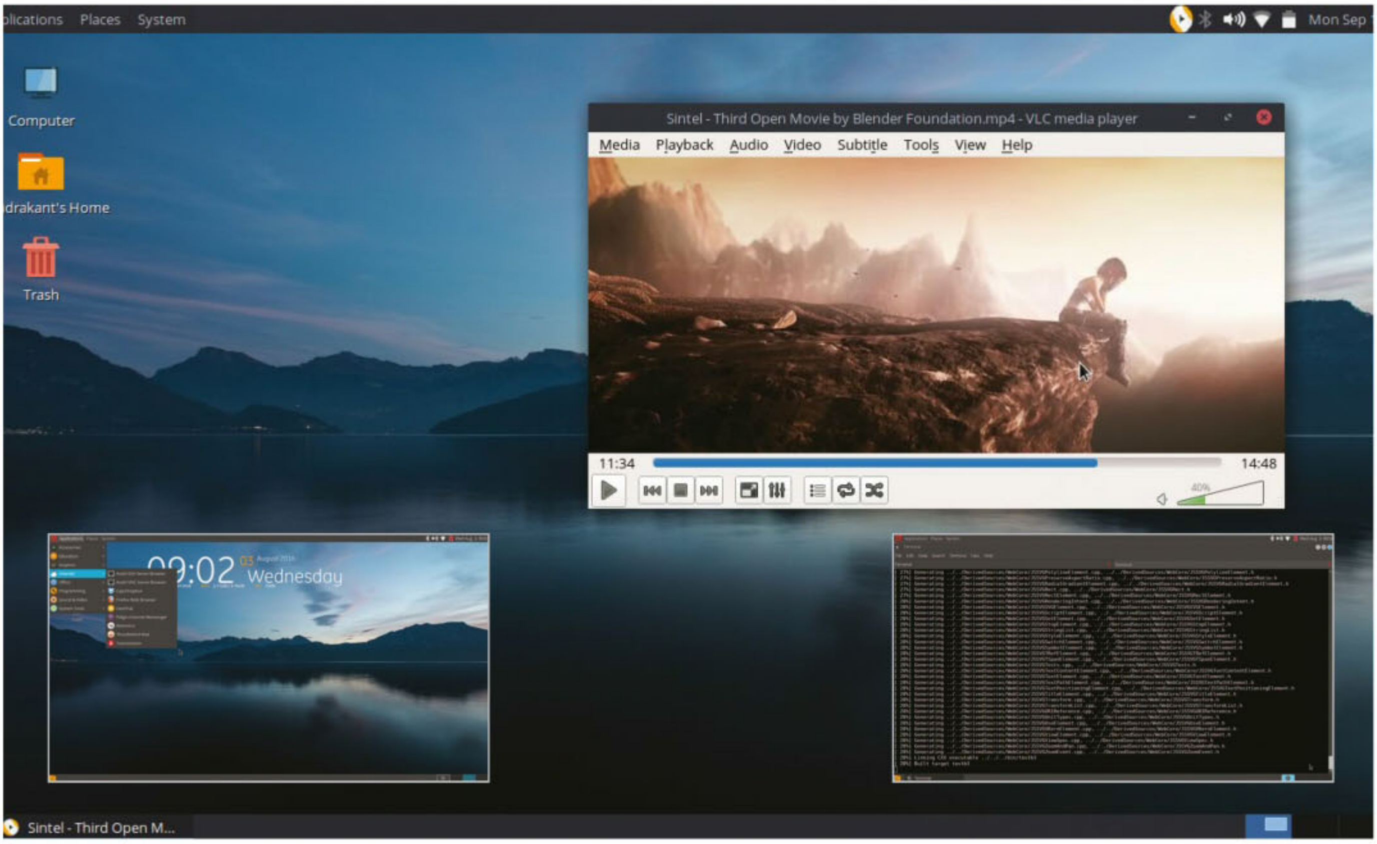
Can a distro usurp Puppy Linux as the king of distros for low-resource machines?

Available from
aryalinux.org

Specs
20GB HD space
1GHz CPU
32-bit platform

When it comes to low-resource machines, finding a distribution that functions well is a surprisingly difficult task. Namely due to the constraints they tend to include, but also the omissions normally in abundance. For a long time, Puppy Linux was regarded as arguably the best low-resource distribution available, but that's somewhat changed in recent times. One of the fast-rising distributions of choice is AryaLinux. Based on the popular LFS (Linux From Scratch), AryaLinux is celebrating its 2017 update, and with some fanfare, we should add.

One of the pleasures of Arya has always been its simplistic installation system, which remains a focal point of the latest build. Thanks largely to user feedback, slowdown issues found in previous versions have all been eradicated, and tailoring the finer points of the desktop to match the limited resources your machine may offer is easier than ever. Similarly to previous versions, Arya comes in two noticeable flavours, MATE and XFCE. The bulk of the improvements lie with the MATE version, with



> There's a real sense of improvement within every update, and it's even more noticeable now

the environment being upgraded to its 1.17.0 version. As you'd expect, it's a pleasure to use throughout and MATE will always be one of the very best desktop environments to use for limited computers.

Another highlight of Arya has been its long-standing package management system, Alps. There's a lot to love about Alps in Arya 2017, which can now be used to build packages via either the source URL or tarball. Not only does this work like a charm, but the fact that all package updates can also be done through Alps as well is a big bonus on top of that. One caveat is Alps' new script for picking up build scripts directly from GitHub; we had some real problems getting this to function on our machine.

Build scripts are also getting a big boost here, to the delight of developers around the world. They can now be used to create ISO files and make stage-wise backups whenever necessary, while it's actually possible to use these scripts to help develop KDE and GNOME desktop environments from source code. We're especially intrigued to see how this feature develops, as it was very rough around the edges when we went hands-on with it.

Going back across to the desktop side of things, Arya again tailors its chosen bundled software to low-powered machines. LibreOffice has been updated to version 5.2.3, while the Parole Media Player and Exaile take centre stage when it comes to media playback. These are two decent choices in their fields, but there are better alternatives to download and they aren't best suited for under-powered units.

Despite finding a lot of good with AryaLinux 2017, we were slightly disappointed to learn that the distribution will only be released in 64-bit going forward. It's a blow to those relying on 32-bit ISO downloads, and as of yet, there's been no communication about the decision to leave behind the 32-bit. It'll certainly leave some users unhappy, but we're sure there's a reasonable explanation behind it.

Looking past the 32-bit conundrum, there's a lot to love about what AryaLinux brings to the table. There's a real sense of improvement within every update, and it's even more noticeable now. Tailoring the distribution is something that every low-resource machine needs, and while the process tends to be difficult, Arya strips it back and oozes simplicity throughout. When you then throw in Alps, which we honestly reckon is one of the best package management systems going, Arya is a winner in our eyes.

Oliver Hill

Pros

A pleasure to use throughout, and Alps is a particular highlight when it comes to package management.

Cons

Certain features still have kinks to be ironed out, and the lack of 32-bit compatibility is a real shame.

Summary

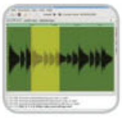
Choices are limited for those with a low-powered machine, and while Puppy is certainly still the best of the distros that cater for these, AryaLinux does bring something new to the table. Big improvements have been made throughout, and while it's far from perfect, it is a great platform for both casual users and budding developers alike

8

SOUND EDITOR

EKO sound editor 5.3.1

There's always a role for simple and quick software

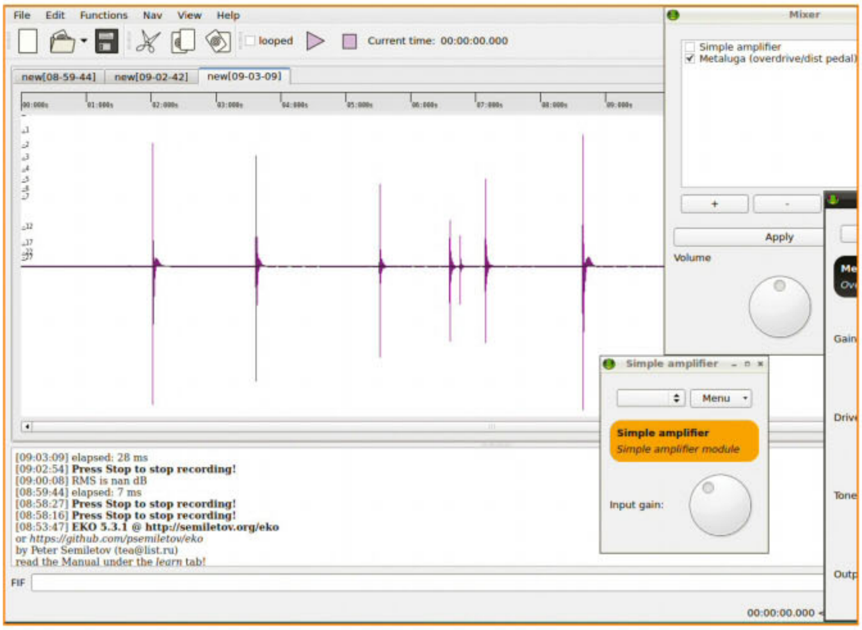


Whether you're running a very old laptop, a small board like the Raspberry Pi, or just want to get a simple job done quickly without reading through a 200-page manual, there's always a need for simple versions of everyday tools. Eko is a sound editor, working with Qt4 or Qt5 and with fairly minimal dependencies, relying on software you may already have installed such as libsndfile.

Written by Peter Semiletov, the author of the Tea editor that we reviewed in **LU&D #159**, this is a newer project, and although far from fully featured, is under fairly active development. Installation went without a hitch. Included in the source tarball is a manual in HTML format - in English and Russian - covering the basics of each feature.

In use, the multiple window layout feels a little over-the-top after the recent trend towards single window interfaces. There's a good case for it though, as the soundfile editing is the main task, and everything else is peripheral. Editing features are minimal, but when you want to do a simple audio task quickly, this is a nice app to have.

The software is licensed under the Public Domain, but there is a link from the author's website for donations to a local Ukrainian dog shelter.



```
[09:03:09] elapsed: 28 ms
[09:02:54] Press Stop to stop recording!
[09:00:08] RMS is nan dB
[08:59:44] elapsed: 7 ms
[08:58:27] Press Stop to stop recording!
[08:58:16] Press Stop to stop recording!
[08:53:47] EKO 5.3.1 @ http://semiletov.org/eko
or https://github.com/psemiletov/eko
by Peter Semiletov (tea@list.ru)
read the Manual under the learn tab!
```

Above Simple capture and simple effects – sometimes quick and easy is better than powerful and complex!

Pros

An easy install, with few dependencies, and a gentle learning curve - simple and reasonably user-friendly.

Cons

Even for a simple editor, it may be lacking your favourite feature. The HTML manual is a little basic.

Great for…

A quick edit done in the time others take to load

semiletov.org/eko/index.html

CPU INFORMATION TOOL

cpuid 20170122

Get detailed CPU information dumped to the command line

You may have run cat /proc/cpuinfo when you weren't sure of the processor(s) on a machine you were logged into, particularly before downloading software compiled for particular chips. It tells you what you need, but there are other occasions where you'll want to know the complete output of the CPUID information that the manufacturer has put onto the chip.

Cpuid is a useful little command line utility for getting that information, and dumping it out to the command line. Pages of it. There's also an option to dump the raw hex info, and the author of the utility is always pleased to get output in this form to help further debug and update cpuid. Updates are regular, bringing in new chip families, and fixing bugs – usually reflecting bugs in the manufacturer's CPUID implementation rather than the tool itself.

Downloading, untarring and running proved no problem on various machines, which is always a good sign (there are many apps that don't make it to these pages because they are nowhere near as co-operative). The included man page gives some insight into the problems of certain chipset families – all x86 relatives, ARM and other processors don't have a direct equivalent for outputting – and provides further reading. As well as a binary tarball, RPM and source packages are available. It's another great little utility in the UNIX spirit of a program just doing one thing, but doing it extremely well.

Pros

As detailed, accurate, and up-to date as the author can make it. Man page included.

Cons

A little opaque to those needing to find the CPU info for the first time. Not ARM.

Great for…

It certainly saves tearing off the heatsink to check!

etallen.com/cpuid.html

BYTECODE TO JAVASCRIPT COMPILER

Js_of_ocaml 2.8.4

OCaml in the browser: powerful, expressive, typesafe code

Perhaps you've come to OCaml through its relative SML, and the wonderful University of Washington MOOC on programming languages; perhaps through curiosity about F#'s parent, or maybe a search for statically typed languages that compile to JavaScript. Perhaps this is the first time you've looked at it. However you've got here, welcome to strong, static typing and a great reduction in runtime errors.

Install OCaml from your operating system's packaging manager and you'll get the OPAM package manager, from which

```
opam install js_of_ocaml
```

will grab you all of the dependencies. However, Js_of_ocaml is a component of the Ocsigen web application framework and

```
opam install ocsigen-start
```

will give you Js_of_ocaml within a wider group of packages.

The website features API documentation and a tutorial on working with Ocsigen. Using the Ocsigen component Eliom, you can write client and server-side code in OCaml, in a single program, with automatic bidirectional communication. The session mechanism allows functional reactive web pages, and quite complex sites take very few lines of code. Because Js_of_ocaml takes OCaml bytecode as its input, you can even use it with OCaml libraries for which you don't have access to the source code. If you've never written OCaml before, don't be afraid of diving in and pushing on through the confusion.

Pros

Expressive language with good record on runtime safety, without the overhead of runtime checks.

Cons

Hindley-Milner type system is a discipline too far for some (try ClojureScript instead)!

Great for…

Maintainable web code from an expressive, functional programming language.

ocsigen.org/js_of_ocaml/

STATIC WEBSITE GENERATOR

Hugo 0.18

Bring website generation back to the text editor!

Hugo is a static blogging solution that has enough flexibility to build many types of website, and features the usual extra like RSS feed generation, and customisable URLs and aliases. It won't be long before you are up and running – you don't even need a Go language compiler, as Hugo is available as a binary which means – as the website boasts – "simply download and run!"

The first task is to set up the site with:

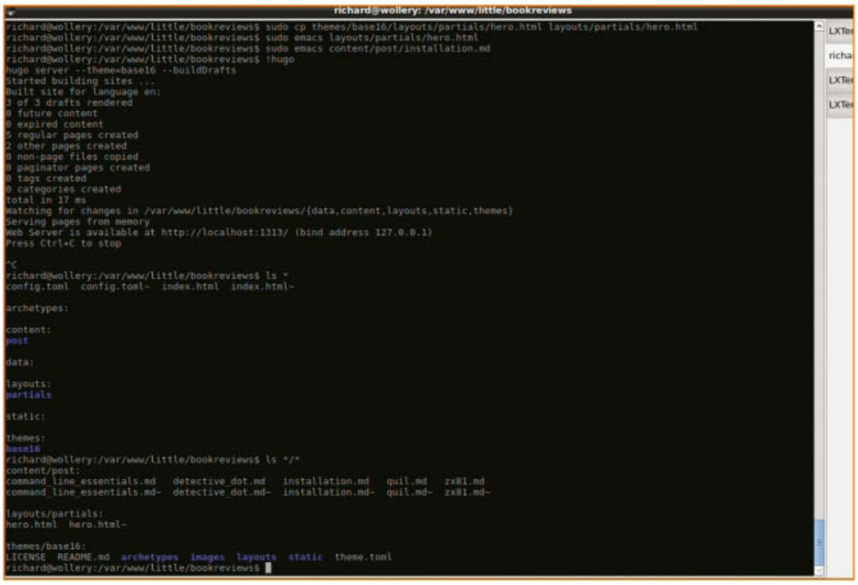
```
hugo new site name-of-site
```

which generates the scaffold for the site. Next:

```
hugo new post/hello-world.md
```

generates a Markdown page for you to edit, creating the first post. Templates use the Go html/template library, enabling you to make quite complex sites. Hugo ships with its own web server for testing, but it's actually good enough for production use if you don't already have an Apache or Nginx setup to hook into. You'll need to install a theme before you can see your site this way.

Alternatively let Hugo compile your posts (typically taking around 1ms per page), and upload the HTML produced to your server anywhere. Documentation is good for getting you up and running quickly and covers areas like the many migration tools available to get your existing site into Hugo.



```
richard@wollery:/var/www/little/bookreviews$ sudo cp themes/base16/layouts/partials/hero.html layouts/partials/hero.html
richard@wollery:/var/www/little/bookreviews$ sudo emacs layouts/partials/hero.html
richard@wollery:/var/www/little/bookreviews$ sudo emacs content/post/installation.md
richard@wollery:/var/www/little/bookreviews$ !hugo
hugo server --theme=base16 --buildDrafts
Started building sites ...
Built site for language en:
3 of 3 drafts rendered
0 future content
0 expired content
5 regular pages created
2 other pages created
0 non-page files copied
0 paginator pages created
0 tags created
0 categories created
total in 17 ms
Watching for changes in /var/www/little/bookreviews/{data,content,layouts,static,themes}
Serving pages from memory
Web Server is available at http://localhost:1313/ (bind address 127.0.0.1)
Press Ctrl+C to stop

^C
richard@wollery:/var/www/little/bookreviews$ ls *
config.toml  config.toml~  index.html  index.html~

archetypes:

content:
post

data:

layouts:
partials

static:

themes:
base16
richard@wollery:/var/www/little/bookreviews$ ls */*
content/post:
command_line_essentials.md   detective_dot.md   installation.md   quil.md   zx81.md
command_line_essentials.md~  detective_dot.md~  installation.md~  quil.md~  zx81.md~

layouts/partials:
hero.html  hero.html~

themes/base16:
LICENSE  README.md  archetypes  images  layouts  static  theme.toml
richard@wollery:/var/www/little/bookreviews$ 
```

Above Everything is a file again - no more delving around in a database obscured by a web interface

Pros

Very fast, flexible and quick to set up. It's great to have sites without database back-ends to administer.

Cons

The command line interface means you won't persuade WordPress-using friends to try it. Comments are not static.

Great for…

Blog sites without the overhead of a blogging engine.

gohugo.io/

RECOMMENDED

Hosting listings

Featured host:

www.cyberhostpro.com
0845 527 9345

CYBERHOSTPRO

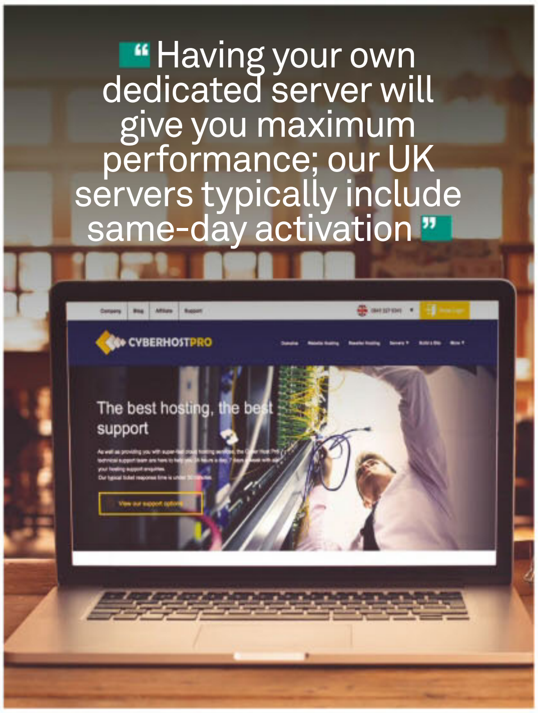
About us

Cyber Host Pro are committed to provide the best cloud server hosting in the UK; we are obsessed with automation and have been since our doors opened 15 years ago! We've grown year on year and love our solid growing customer base who trust us to keep their business's cloud online!

If you're looking for a hosting provider who will provide you with the quality you need to help your business grow then contact us to see how we can help you and your business! We've got a vast range of hosting solutions including reseller hosting and server products for all business sizes.

What we offer

- Cloud VPS Servers – scalable cloud servers with optional Cpanel or Plesk control panel.
- Reseller Hosting – sell web and email hosting to your clients; both Windows and Linux hosting available.
- Dedicated Servers – having your own dedicated server will give you maximum performance; our UK servers typically include same-day activation.
- Website Hosting – all of our web hosting plans host on 2015/16 SSD Dell servers giving you the fastest hosting available!



“Having your own dedicated server will give you maximum performance; our UK servers typically include same-day activation”

5 Tips from the pros

01 Optimise your website images
When uploading your website to the internet, make sure all of your images are optimised for websites! Try using **jpegmini.com** software, or if using Wordpress install the EWWW Image Optimizer plugin.

02 Host your website in the UK
Make sure your website is hosted in the UK, not just for legal reasons! If your server is overseas you may be missing out on search engine rankings on **google.co.uk** – you can check where your site is on **www.check-host.net.**

03 Do you make regular backups?
How would it affect your business if you lost your website today? It is important to always make your own backups; even if your host offers you a backup solution it's important to take responsibility for your own data.

04 Trying to rank on Google?
Google made some changes in 2015. If you're struggling to rank on Google, make sure that your website is mobile-responsive! Plus, Google now prefers secure (https) websites! Contact your host to set up and force https on your website.

05 Avoid cheap hosting
We're sure you've seen those TV adverts for domain and hosting for £1! Think about the logic… for £1, how many clients will be jam-packed onto that server? Surely they would use cheap £20 drives rather than £1k+ enterprise SSDs! Try to remember that you do get what you pay for!

Testimonials

Chris Michael
“I've been using Cyber Host Pro to host various servers for the last 12 years. The customer support is excellent, they are very reliable and great value for money! I highly recommend them.”

Glen Wheeler
“I am a website developer, I signed up with Cyber Host Pro 12 years ago as a small reseller, 12 years later I have multiple dedicated and cloud servers with Cyber Host Pro, their technical support is excellent and I typically get 99.9-100% uptime each month”

Paul Cunningham
“Me and my business partner have previously had a reseller account with Cyber Host Pro for 5 years, we've now outgrown our reseller plan, Cyber Host Pro migrated us to our own cloud server without any downtime to our clients! The support provided to us is excellent, a typical ticket is replied to within 5-10 minutes! ”

Supreme hosting

CWCS MANAGED HOSTING SUPREME HOSTING. SUPREME SUPPORT.

www.cwcs.co.uk
0800 1 777 000

CWCS Managed Hosting is the UK's leading hosting specialist. They offer a fully comprehensive range of hosting products, services and support. Their highly trained staff are not only hosting experts, they're also committed to delivering a great customer experience and passionate about what they do.

- Colocation hosting
- VPS
- 100% Network uptime

Value hosting

elastichosts

elastichosts.co.uk
02071 838250

ElasticHosts offers simple, flexible and cost-effective cloud services with high performance, availability and scalability for businesses worldwide. Their team of engineers provide excellent support around the clock over the phone, email and ticketing system.

- Cloud servers on any OS
- Linux OS containers
- World-class 24/7 support

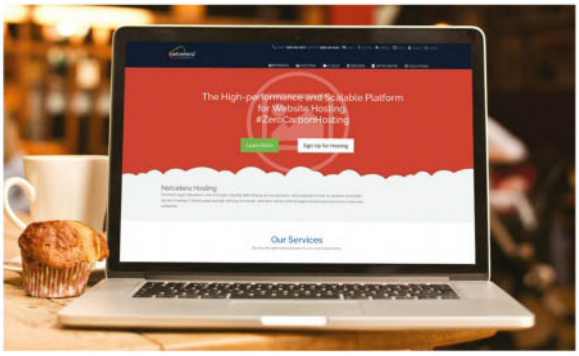
Small business host

HOSTPAPA

www.hostpapa.co.uk
0800 051 7126

HostPapa is an award-winning web hosting service and a leader in green hosting. They offer one of the most fully featured hosting packages on the market, along with 24/7 customer support, learning resources, as well as outstanding reliability.

- Website builder
- Budget prices
- Unlimited databases


The High-performance and Scalable Platform for Website Hosting #ZeroCarbonHosting
Our Services

Enterprise hosting:

netcetera

www.netcetera.co.uk | 0800 808 5450

Formed in 1996, Netcetera is one of Europe's leading web hosting service providers, with customers in over 75 countries worldwide. As the premier provider of data centre colocation, cloud hosting, dedicated servers and managed web hosting services in the UK, Netcetera offers an array of services to effectively manage IT infrastructures. A state-of-the-art data centre enables Netcetera to offer your business enterprise-level solutions.

- Managed and cloud hosting
- Data centre colocation
- Dedicated servers

Budget hosting:

HETZNER ONLINE

www.hetzner.de/us | +49 (0)9831 5050

Hetzner Online is a professional web hosting provider and experienced data centre operator. Since 1997 the company has provided private and business clients with high-performance hosting products as well as the necessary infrastructure for the efficient operation of websites. A combination of stable technology, attractive pricing and flexible support and services has enabled Hetzner Online to continuously strengthen its market position both nationally and internationally.

- Dedicated and shared hosting
- Colocation racks
- Internet domains and SSL certificates
- Storage boxes

SSD Web hosting

bargainhost

www.bargainhost.co.uk
0843 289 2681

Since 2001 Bargain Host have campaigned to offer the lowest possible priced hosting in the UK. They have achieved this goal successfully and built up a large client database which includes many repeat customers. They have also won several awards for providing an outstanding hosting service.

- Shared hosting
- Cloud servers
- Domain names

Value Linux hosting

PATCHMAN WEB HOSTING

patchman-hosting.co.uk
01642 424 237

Linux hosting is a great solution for home users, business users and web designers looking for cost-effective and powerful hosting. Whether you are building a single-page portfolio, or you are running a database-driven ecommerce website, there is a Linux hosting solution for you.

- Student hosting deals
- Site designer
- Domain names

Fast, reliable hosting

:BYTEMARK

www.bytemark.co.uk
01904 890 890

Founded in 2002, Bytemark are "the UK experts in cloud & dedicated hosting". Their manifesto includes in-house expertise, transparent pricing, free software support, keeping promises made by support staff and top-quality hosting hardware at fair prices.

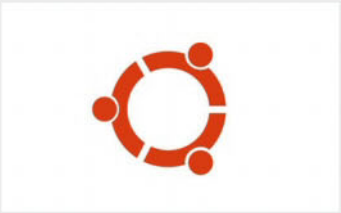
- Managed hosting
- UK cloud hosting
- Linux hosting

Free with your magazine

Instant access to these incredible free gifts…

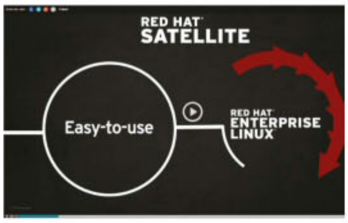
The best distros and FOSS

Essential software for your Linux PC



Professional video tutorials

The Linux Foundation shares its skills



Tutorial project files

All the assets you'll need to follow our tutorials

Plus, all of this is yours too…

- Download five specialist distros for working with containers
- Easy access to the essential tools you need, such as Docker
- Enjoy 20 hours of expert video tutorials from The Linux Foundation
- Get the program code for our Linux and Raspberry Pi tutorials

Log in to **www.filesilo.co.uk/linuxuser**

Register to get **instant access** to this pack of must-have Linux distros and software, how-to videos and tutorial assets

Free for digital readers too!
Read on your tablet, download on your computer

FREE 5 LIGHTWEIGHT DISTROS
www.linuxuser.co.uk
Linux User & Developer
THE ESSENTIAL MAGAZINE FOR THE GNU GENERATION
WORK FASTER, MORE SECURELY, FROM ANYWHERE
TAKE CONTROL OF CONTAINERS
Run Linux on any machine • Deploy software on demand
ESSENTIAL LINUX FIXES
RESTRICT UNTRUSTED SOFTWARE WITH FIREJAIL
Reduce security risks by running apps in a sandbox
KEEP YOUR DATA SAFE
RUN A VPN
FileSilo.co.uk
LinuxUser
#all
HOME
FAQs
SUBSCRIBERS
FILTER CONTENT BY TYPE
Show All
Free Stuff
3D Assets
Software
Tutorial Files
Video Tutorials
Distros
More
Most Downloaded Files
Linux User & Developer 167
Linux User & Developer 166
Linux User & Developer 165
Linux User & Developer 164
Highest Rated Files
MacBook Air

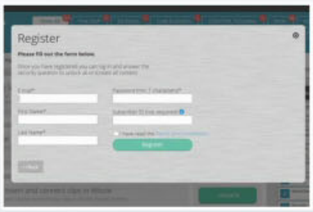
FileSilo

The home of great downloads – exclusive to your favourite magazines from Future Publishing

- **Secure and safe online access, from anywhere**
- **Free access for every reader, print and digital**
- **Download only the files you want, when you want**
- **All your gifts, from all your issues, in one place**

Get started

Everything you need to know about accessing **your FileSilo account**



01 Follow the instructions on screen to create an account with our secure FileSilo system. Log in and unlock the issue by answering a simple question about the magazine.

02 You can access FileSilo on any computer, tablet or smartphone device using any popular browser. However, we recommend that you use a computer to download content, as you may not be able to download files to other devices.

03 If you have any problems with accessing content on FileSilo take a look at the FAQs online or email our team at the address below

filesilohelp@imagine-publishing.co.uk

An incredible gift for subscribers

Subscribe today & unlock the free gifts from more than 40 issues

Access our entire library of resources with a money saving subscription to the magazine – that's hundreds of free resources

Over 20 hours of video guides

Essential advice from the Linux Foundation

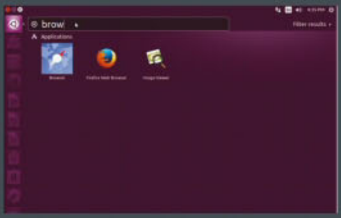
The best Linux distros

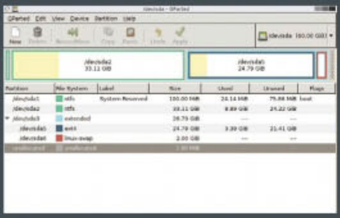
Specialist Linux operating systems

Free Open Source Software

Must-have programs for your Linux PC

```
$ cat /proc/sys/kernel/perf_event_paranoid
1
$ su -
Password:
thinkpad ~ # echo 0 > /proc/sys/kernel/perf_event_paranoid
thinkpad ~ # exit
logout
$ ls /etc/sysctl.conf
/etc/sysctl.conf
$ cd ~
$
```





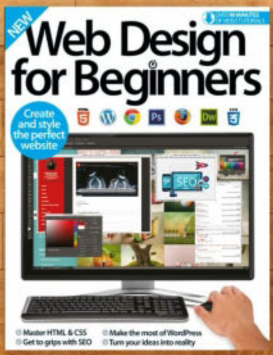
Head to page 32 to subscribe now

Already a print subscriber?
Here's how to unlock FileSilo today…

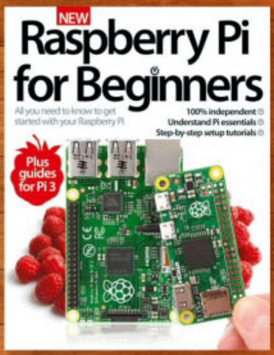
Unlock the entire LU&D FileSilo library with your unique Web ID – the eight-digit alphanumeric code that is printed above your address details on the mailing label of your subscription copies. It can also be found on any renewal letters.

More than 400 reasons to subscribe

More added every issue



NEW
Windows 10 Tips, Tricks & Apps
Unlock the full potential of Windows 10
Includes all the latest updates



Discover another of our great bookazines

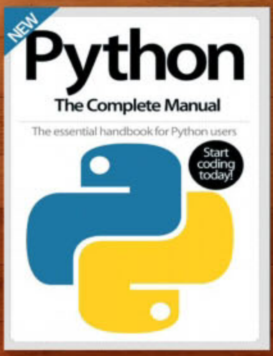
And save 20% on your first order when you buy direct from us

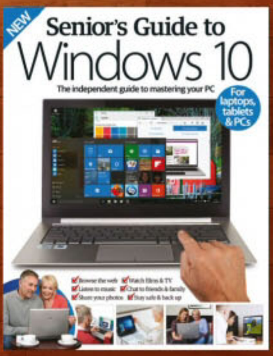
MAKE SCRATCH VIDEOGAMES
100% UNOFFICIAL
RasPi FOR KIDS
DISCOVER THE POWER OF PI!
THE ULTIMATE GUIDE TO CODING AND BUILDING WITH YOUR FIRST PI
BUILD ROBOTS AND GADGETS!

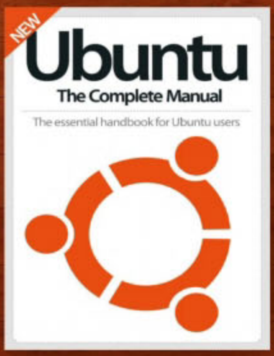
BRAND NEW!

Raspberry Pi for Kids

Raspberry Pi is a great learning tool, helping to introduce your kids to the world of coding and programming. This book is packed with fun projects!


NEW
Python The Complete Manual
The essential handbook for Python users
Start coding today!




NEW
Ubuntu The Complete Manual
The essential handbook for Ubuntu users

✓ Get great savings when you buy direct from us

✓ 1000s of great titles, many not available anywhere else

✓ World-wide delivery and super-safe ordering

Future **www.imagineshop.co.uk**
Magazines, back issues & bookazines.

Use the code "WELCOME20" when you make your first order to receive your discount

LOSE YOURSELF IN A WORLD OF
Vinyl
FIND YOURSELF IN OXFAM'S ONLINE SHOP
oxfam.org.uk/shop
OTO
COLUMBIA
TIME ON MY HANDS
OXFAM

HETZNER ONLINE

Network security

Firewall.

Rules
(incoming)

	Name	Source IP	Destination IP	Source port	Destina		flags	Action
#1	icmp	0.0.0.0/0	0.0.0.0/0	0-65535	0-65535	icmp	syn|ack|fin|rst|psh|urg	accept
#2	ssh	85.10.212.62/32	0.0.0.0/0	0-65535	22		n|rst|psh|urg	accept
#3	smtp	0.0.0.0/0					|rst|psh|urg	accept
#4	http	0.0.0.0/					fin|rst|psh|urg	accept
#5	pop3	0.0.					|rst|psh|urg	accept
#6	imap	0.0.					fin|rst|psh|urg	accept
#7	tcp established	0.0.						accept

Add rule

intel XEON inside

e.g. Dedicated Root Server PX61-NVMe

Intel® Xeon® E3-1275 v5
Quad-Core Skylake Processor
64 GB DDR4 ECC RAM
2 x 512 GB NVMe Gen3 x4 SSD
Guaranteed 1 Gbit/s bandwidth
100 GB Backup Space
30 TB traffic inclusive*
No minimum contract
Setup Fee £101.00

monthly £ **50**

Free Firewall for Your Dedicated Root Servers!

Hetzner Online's stateless firewall is a free security solution for your dedicated root server. Starting now on the customer interface Robot, you can use the firewall feature to define your own filtering settings for traffic, such as the originating IPv4 address or TCP/UDP sender port. With this feature, Hetzner Online helps you protect your dedicated root server from Internet dangers. **And it is naturally free of cost.**

www.hetzner.de/gb

* There are no charges for overage. We will permanently restrict the connection speed if more than 30 TB/month are used. Optionally, the limit can be permanently cancelled by committing to pay £1 per additional TB used.

All prices exclude VAT and are subject to the terms and conditions of Hetzner Online GmbH. Prices are subject to change. All rights reserved by the respective manufacturers. Intel, Intel Logo, Intel Xeon and Xeon Inside are brands of the Intel Corporation in the USA or other countries.