NEW

Genius Guide

OVER 6 HOURS OF VIDEO TUTORIALS

Faster, better servers now!

Linux & Open Source

The essential guide to mastering open source software and operating systems

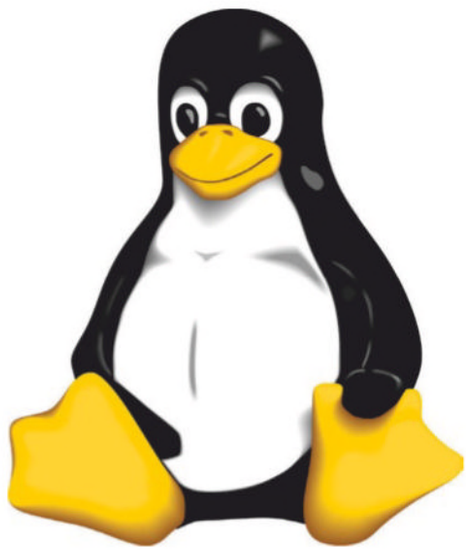
Ultimate distro guide • Master Ubuntu • Triple boot your system

Genius Guide

A comprehensive masterclass to becoming an instant expert

Welcome to Linux & Open Source

In an age where most people are carrying a fully-functioning computer and matching operating system in their pocket (if not on their wrist, with the advent of wearable technology), it's easy to forget the fun that can be had with editing and customising your own system. Linux and other open source software packages can give you an exhilarating sense of freedom in making your computer your own, no matter what you use it for. In this Genius Guide, you'll learn advanced tips for how to get the most out of the latest distros, and find projects to try out. We reveal how to get faster, better servers, and show you how to triple boot your sytem. Whether you're looking for useful hacks like using the web from the terminal, or want something more technical like signal handling or code analysis, we have everything you need to become a Linux and open source expert in no time at all.



Genius Guide

Linux & Open Source

Imagine Publishing Ltd
Richmond House
33 Richmond Hill
Bournemouth
Dorset BH2 6EZ
☎ +44 (0) 1202 586200
Website: www.imagine-publishing.co.uk
Twitter: @Books_Imagine
Facebook: www.facebook.com/ImagineBookazines

Publishing Director
Aaron Asadi

Head of Design
Ross Andrews

Production Editor
Hannah Westlake

Senior Art Editor
Greg Whitaker

Designer
Anne-Claire Pickard

Printed by
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
Tel 0203 787 9060 www.marketforce.co.uk

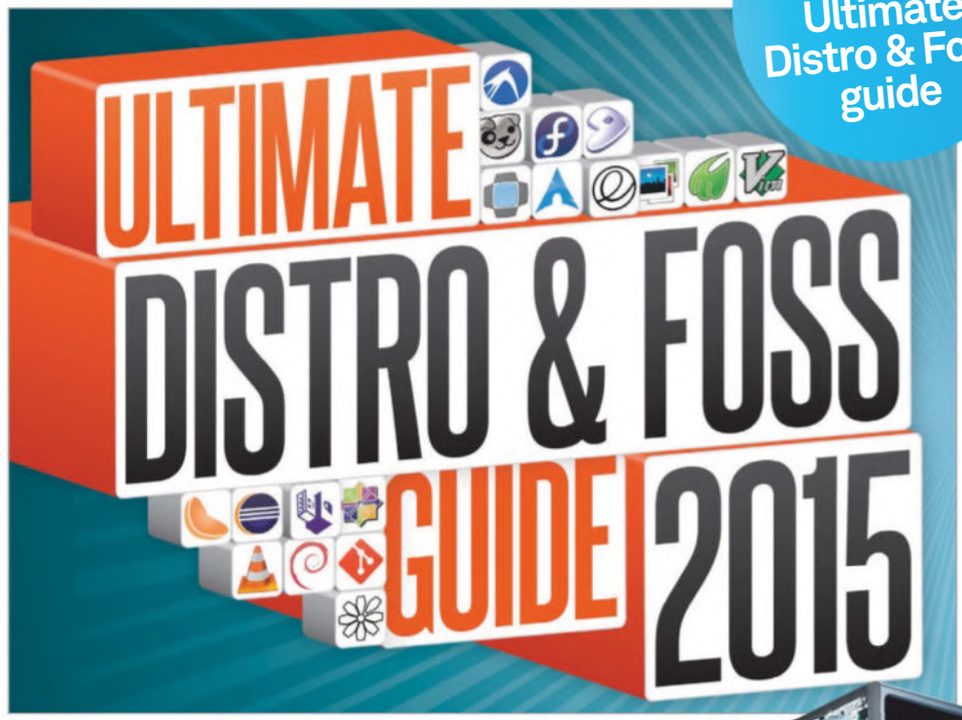
Distributed in Australia by
Network Services (a division of Bauer Media Group), Level 21 Civic Tower, 66-68 Goulburn Street, Sydney, New South Wales 2000, Australia Tel +61 2 8667 5288

ISBN: 978 1 78546 205 4

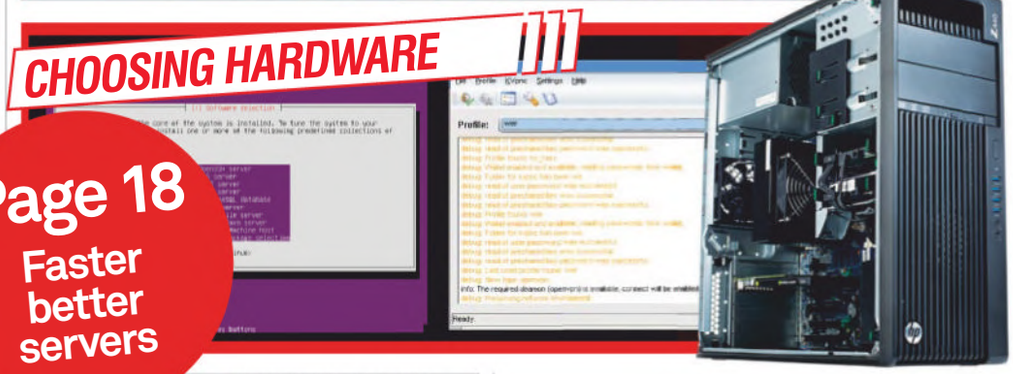
Part of the LinuxUser & Developer bookazine series

Contents

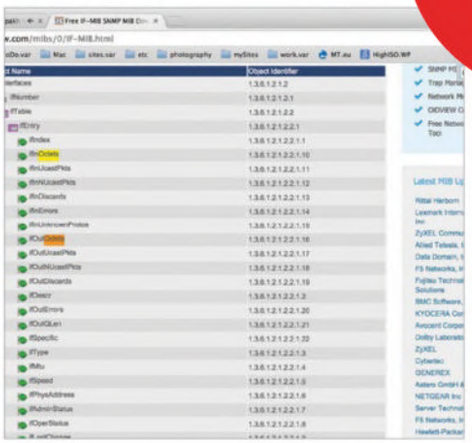
Your guide to what's inside

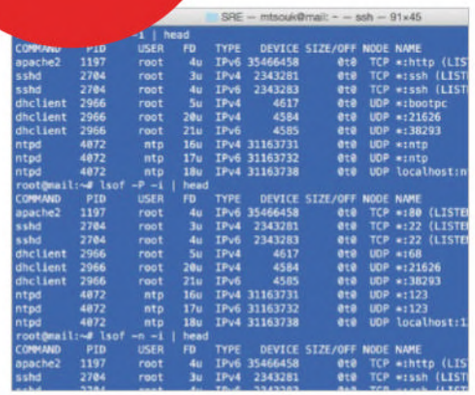


Page 8
Ultimate Distro & Foss guide

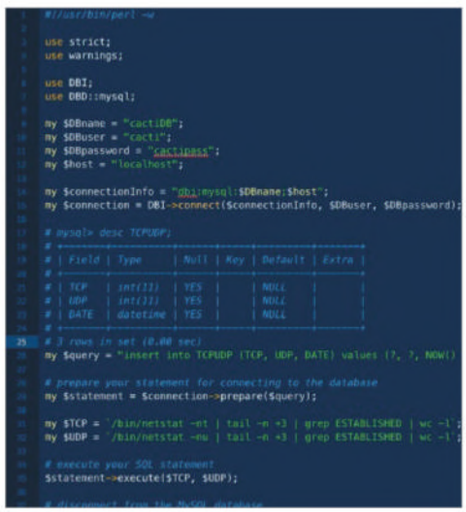


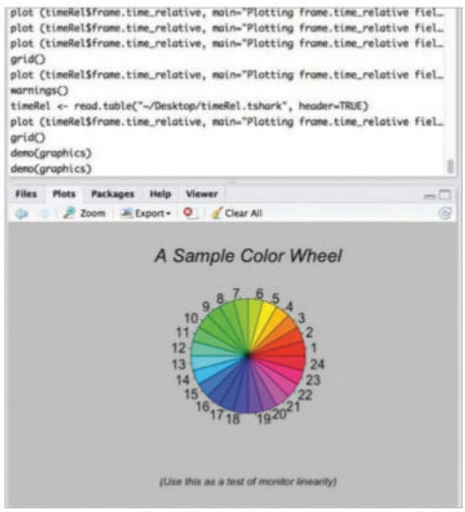
Page 18
Faster better servers



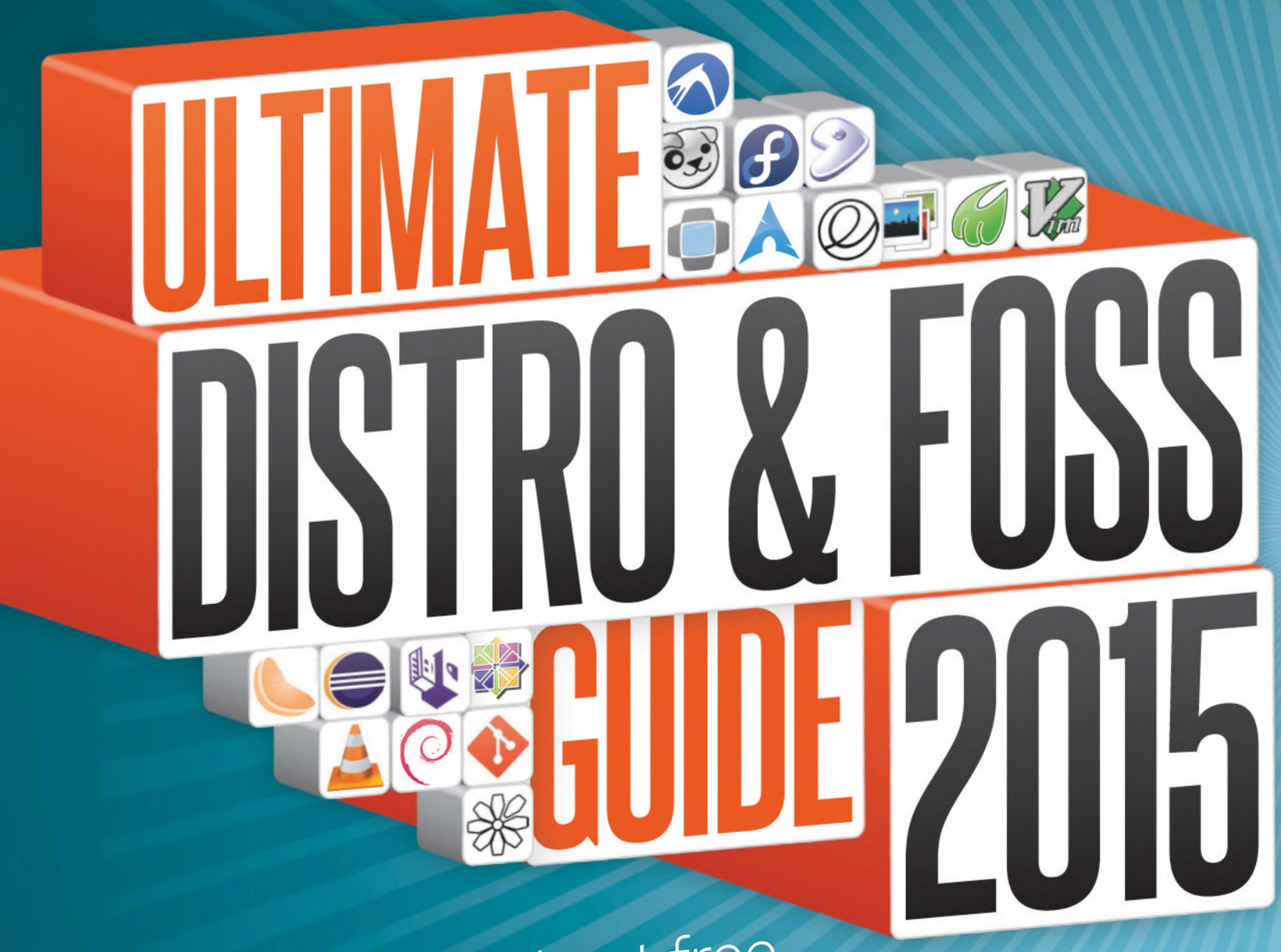


Developer guide





"An everyday distro is quite a broad thing; in this context we mean the kind of OS you can use for just about anything"



Discover the best free software in every category and see what you should be using this year

BEST DISTRO FOR...

Everyday

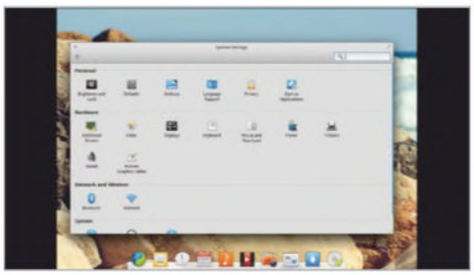
Is this the year of Linux? Or is it the year we stop claiming that? Either way, there are already plenty of choices for your day-to-day

Elementary OS

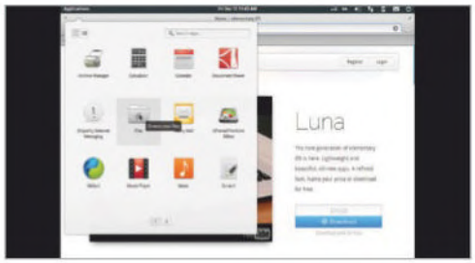
An everyday distro is quite a broad thing; in this context we mean the kind of operating system you can use for just about anything and everything without really specialising in one specific area. Something that's easy to use and its supplied tools aid your use of it.

This is exactly the point where elementaryOS comes in. Aiming to be easy to use for people of all skill levels, elementaryOS is a beautifully designed distro that has had a lot of care put into it. Using an Ubuntu LTS as a base and cribbing from a lot of existing design decisions, elementaryOS is hardly a completely original Linux distribution.

What makes elementaryOS unique is its use of these design aspects and design decisions, putting together a wholly new desktop and distro experience



Elements like the dock and window styling will look familiar



The simple, searchable applications menu takes design cues from mobile operating systems

that you can't find anywhere else without some serious customisations on the user's part. It's the best of every world for people who prefer using a fully-feature graphical desktop, and it works extremely well on new and modern systems.

The wording on the website is inclusive and friendly to newcomers as well – not once is there mention of Linux or distribution, instead using wording familiar to everyone and rightfully referring to elementaryOS as a whole as an operating system. This kind of friendliness and familiarity is translated to the desktop, from a simple dock bar that grants access to important programs from the moment you start using it to an applications menu reminiscent of modern smartphone design.

The stable Ubuntu base also grants access to an unprecedented level of packages and other desktop types if you want something a little different to elementary's offering. It's a great first distro for people who want to make the switch to Linux as well.

ALTERNATIVE DISTROS

Linux Mint

Another firm favourite as an everyday distro, Linux Mint would have taken this category by storm once upon a time due to excellent design over two fantastic desktops on top of an excellent distro.

Ubuntu

Ubuntu is probably the most popular distro in the world, or at least the most well-known, which means a lot of software supports it and not other distros. You can customise it anyway you want off of its core base.

Mageia

Mageia is a very user-friendly spin on the Mandriva family with some excellent apps for controlling just about every aspect of the distro along with other smart design choices. It's been brought back into Mandriva but is still great.

DESKTOP ENVIRONMENT	9/10
DOCUMENTATION	7/10
CUSTOMISATION	6/10
COMMUNITY	8/10
PACKAGES	10/10
PERFORMANCE	9/10
INSTALLATION	9/10
HARDWARE SUPPORT	7/10

BEST FOSS

LibreOffice

The office suite that has far superseded its originator, LibreOffice can handle all your word processing, spreadsheeting and presentation needs extremely well with a selection of excellent software.

Firefox

Once again the king of the browsers, with half a billion users around the world, Firefox has privacy and customisability in mind with its design. Due to some excellent cross-platform tools, you can use it wherever you want.

Thunderbird

The email counterpart to Firefox has remained a very strong email client on any operating system for a long time. With a great range of add-ons and extensions, you can have it work exactly as you'd want it to.

Cinnamon

The desktop environment originally made for Linux Mint, Cinnamon uses a more traditional desktop layout and a lot of common sense design choices and workflow methods that make the most of modern tech and traditional ideas. It's an improvement on many default desktops.

Shotwell

Excellent photo management software used by a lot of distros by default, it even has some basic support for RAWs. You can perform batch operations to tweak colours and lighting, or just organise photos into specific tags.

BEST DISTRO FOR...

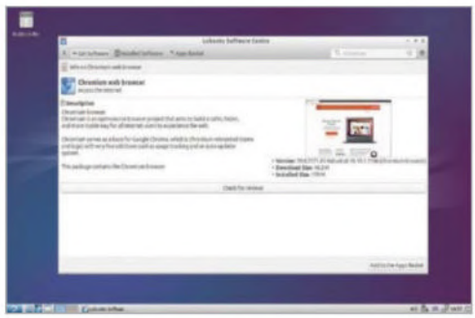
Lightweight

A lighter Linux distro can help you get the most out of an older or underpowered system by using relatively fewer resources

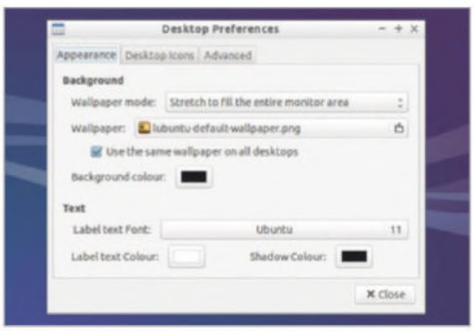
Lubuntu

You can define lightweight in a number of ways these days. While the graphical software part of a distro itself can be the most resource intensive, the core kernel and behind-the-scenes packages can also take power away from CPU cycles. While Lubuntu and other normal Linux distros running LXDE don't do much to the core of their operation, merely running the desktop itself can be a huge relief on some systems.

That's why we're awarding Lubuntu this prestigious prize. While we're not the biggest fans of Unity, the distribution underneath the desktop environment is an incredibly solid and relatively easy-to-use system that a good desktop environment can really make the most of. LXDE is not only extremely light – it uses 78 MB of RAM compared to XFCEs 89 MB – it's also (very importantly) as fully-featured as most modern desktops. Add this to Ubuntu's impressive software and packages and you've got a lightweight distro that doesn't sacrifice any usability and still has access to all your favourite software.



Lubuntu has an excellent Software Centre to make package downloads easier



While minimal, the LXDE desktop styling is very smart

As for lightweight software, Lubuntu comes with quite a different selection of default apps compared to its vanilla counterpart. A smattering of the basics such as Firefox, Pidgin and Abiword are all you're presented with. It's enough to get you started and thanks to access to the full Ubuntu repos you can then start building up your system with some of the excellent and lightweight apps that Linux is known for.

It's definitely not the lightest distro around but it's certainly the best distro that comes under the lightweight banner.

ALTERNATIVE DISTROS

wattOS

A very lightweight and speedy operating system that aims to do two things: boot to desktop very fast and also save you electricity, either plugged in or on the battery. It does both of these exceedingly well.

Puppy Linux

Puppy Linux lets you teach an old dog new tricks – it's specifically designed for older systems and is extremely resource friendly. It can live in a very small amount of RAM yet still includes a very functional system.

Porteus

A very fast live distro that takes up only 300 MB of space, and is optimised to run from live media as well. You can add modules for extra software if needed, making it a very customisable distro.

DESKTOP ENVIRONMENT	8/10
DOCUMENTATION	7/10
CUSTOMISATION	9/10
COMMUNITY	6/10
PACKAGES	10/10
PERFORMANCE	10/10
INSTALLATION	9/10
HARDWARE SUPPORT	9/10

BEST FOSS

Audacious

Very lightweight and very fast, Audacious is the definitely the best audio player for those on a resource budget. It hooks into notification centres of most major desktops as well, allowing you to control it better.

Midori

Midori teeters on the edge of being just a bit too lightweight to be as useful as some of its peers, but it managed to maintain a number of excellent features to make browsing the Internet with it acceptable in 2015.

CMPlayer

A lightweight video player that still has a fairly decent interface and no need for mucking around in the command line, it will play all the media you need as long as you have the right codecs and backends installed.

Geany

A text editor with IDE features that is popular among those with a few small projects on the go. It's easy enough to switch between the two types, meaning you can use it for your day-to-day text editing before going full developer.

Enlightenment

A window manager or full desktop environment, Enlightenment is an incredible flexible and lightweight framework loved by hardcore users. It's rarely used as a default desktop, but give it a go if you're on the hunt for something different.

BEST DISTRO FOR...

Entertainment

Here are your best distros for making custom systems to play all your media, either on a TV or just for better navigation while at your desk

OpenELEC

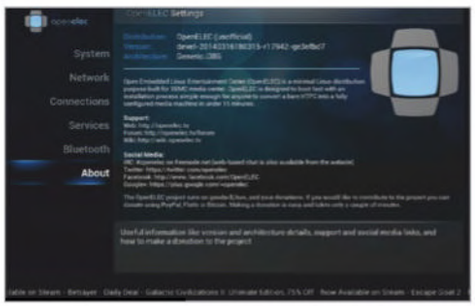
When it comes to media and other forms of entertainment consumption via computers, one of the most recognisable names in existence is XBMC. Very likely because of its quite redundant legacy naming, it's unfortunately dropping the well-known branding in favour of new name Kodi. What's this all got to do with entertainment distros though?

Well the devs behind Kodi also make the excellent OpenELEC, a Linux distro optimised for a number of different hardware types to offer the best possible Kodi experience. Not only does it work on specialist hardware such as the Raspberry Pi, Apple TV and some other hardware ideal as HTPCs, but you can also get generic PC builds for x86 and x64 systems.

OpenELEC takes up only a tiny amount of your file system while still offering a full version of Kodi. This means you can use any remaining storage to keep media local, such as your party playlist, if that's your kind of thing, however you can still stream your media from any other computer set up to share it. Setting up these shared folders in Kodi is quick and simple, and it even scrapes together information on the files to make navigation easier.

There are also plenty of add-ons that allow you to stream media for many online video sources, as well as recently-added native Live TV viewing and recording – the latter being a better use for your hard drive space.

For both TV and PC, this is an excellent way to consume media in almost any situation, with plenty of codecs and online features.



OpenELEC claims to set up your media box in 15 minutes

There are some customisations you can make, as well as add-ons to install

ALTERNATIVE DISTROS

GeeXbox

A direct alternative to OpenELEC that gives you a little more choice for how you set up a HTPC. It hasn't had any new development for about a year now, but that's nothing to worry about as it's still excellent.

Ubuntu

When it comes to being an entertainment distro, Ubuntu's strength is its supreme list of packages. You can set up XBMC/Kodi, Plex, Myth or just plain video and music players on Ubuntu of any type.

AVLinux

Not for playing back your media per se, AV Linux is an excellent way for you to actually create audio and video yourself, thanks to a custom kernel and great package selection on a live CD or live USB stick.

DESKTOP ENVIRONMENT	10/10
DOCUMENTATION	8/10
CUSTOMISATION	4/10
COMMUNITY	8/10
PACKAGES	3/10
PERFORMANCE	10/10
INSTALLATION	9/10
HARDWARE SUPPORT	10/10

BEST FOSS

Kodi

Previously XBMC, Kodi Entertainment Center is the premier media PC software around. It's the software behind OpenELEC and it can be used for simple music and video playback, or streaming services and recording live TV.

VLC

An extremely powerful yet small piece of software that can not only play just about any form of media, but also send, receive and record network streams. It's very customisable and easy to use even if you don't want to stream your desktop.

Clementine

The most fully-featured audio player around, with incredible library and playlist management and an excellent interface to boot. It also has a smart playlist that will build itself on the fly, however it doesn't run well on older systems.

Nuvola

One of the problems we have found with browser-based streaming is that we cannot control playback with media keys or hotkeys. Nuvola allows you to keep all your streaming audio in one place and, more importantly, control it.

GIMP

The powerful image manipulator that is probably the best open source has to offer, GIMP can even challenge Photoshop thanks to its array of excellent features and tools – it even has a more straightforward naming convention in places.

BEST DISTRO FOR...

Development

With Linux proving a popular platform for development work, which is the best of the bunch for getting your code on?

Arch Linux

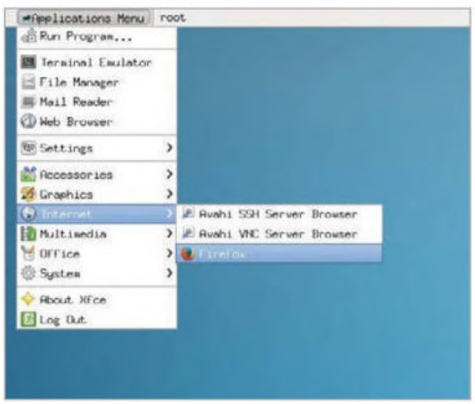
Arch has never been a distribution to pander to the common denominator. While its contemporaries add user-friendly wizards and hand-holding installation packages, Arch dumps the newcomer to a console session and leaves them adrift with little more than a Wiki page for company. For beginners, simply getting Arch installed can seem like a major achievement – but beginners are most certainly not Arch's target market.

Once it's installed, Arch reveals its true potential. It allows the more technical user to install only the packages required for day-to-day work for guaranteed zero bloat, with all the benefits to performance, stability and your ability to concentrate that this implies. Tweaking your Arch install can become obsessive, but once it's set up to your liking you can expect a smooth ride.

Arch is certainly not a distribution for beginners, but those with Linux experience will find plenty to like about it. It has an active community, albeit one which can be unwelcoming to beginners, boasts a great package selection for even some of the more esoteric tools in a developer's arsenal, and promises to provide an easily-customised environment tailored specifically to your individual needs.

A rolling-release development methodology means that while installation may be painful it's a one-off experience, and users are guaranteed to be working on the latest available tools and resources. There's a good reason Arch and its derivatives are popular among Linux kernel developers as well as those who write software for other platforms.

Finally, Arch has an ace up its sleeve for those targeting Arch itself with their creations: the Arch Build System. Designed specifically for Linux developers, the ABS offers the ability to create, customise and distribute packages into Arch which are built directly from source. Based heavily on the BSD ports system, ABS offers automation for tasks other distributions require developers to perform by hand.



You can install just the tools you need for zero bloat

ALTERNATIVE DISTROS

CrunchBang

A Debian derivative, CrunchBang uses the lightweight Openbox window manager to be as distraction-free as possible and offers a good balance between the performance and flexibility of Arch and the shallow learning curve of Ubuntu.

Gentoo

Like Arch, Gentoo features a BSD ports-like package management system dubbed Portage, and a release system ensuring users install the latest packages compiled from source with per-distro optimisations.

Ubuntu

Often derided for Canonical's treatment of the wider open-source community, Ubuntu nevertheless promises wide compatibility backed by the option of commercial support contracts.

DESKTOP ENVIRONMENT	2/10
DOCUMENTATION	5/10
CUSTOMISATION	10/10
COMMUNITY	6/10
PACKAGES	8/10
PERFORMANCE	9/10
INSTALLATION	2/10
HARDWARE SUPPORT	9/10

BEST FOSS

Eclipse

It might lack compatibility with the GNU General Public Licence, but the Eclipse Public Licensed Eclipse IDE is a powerful tool. Based on IBM's Visual Age, it supports most common programming languages you'll be working with.

VirtualBox

While the GPL-licensed VirtualBox OSE build only provides virtualised USB 1.1 support, its other features make it a great way to run alternative operating systems on top of userspace Linux; ideal for testing your code on other platforms.

Git

Born of a copyright confusion that surrounded BitKeeper, Git is the distributed revision control system of choice for kernel developers. It allows for easy collaborative working with plenty of ways to track bugs added in later code revisions.

Vim/EMACS

Did you really think we were going to get involved in this debate? A good text editor is the programmer's best tool, but we're staying on the fence with this one. Whether you're an acolyte of Stallman and Steele or a proselyte for Moolenaar, use whichever of these works for you.

GNU Debugger

The standard debugger for GNU/Linux, GDB's capabilities extend beyond the obvious with support for programming languages ranging from Free Pascal and Ada through to Objective-C and Java. We recommend giving it a try.

BEST DISTRO FOR...

Enterprise

For companies, Linux can significantly reduce the total cost of IT infrastructure, but which distribution stands out?

SUSE Linux Enterprise Desktop/Server

When it comes to desktop Linux distributions for the enterprise crowd, there are two names that go toe-to-toe: SUSE and Red Hat. Both offer distributions for the desktop and server specifically marketed as 'Enterprise Linux', and both back up their offerings with a wealth of commercial support.

With customers as varied as the London Stock Exchange and Office Depot, SUSE Linux Enterprise is extremely popular. Features like SUSE Manager, which provides automation of server management, and SUSE Cloud, providing OpenStack-powered local cloud infrastructure, make it easy to see why.

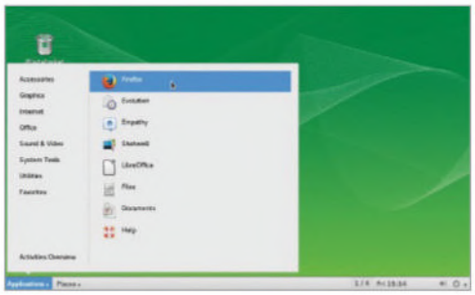
Enterprise users typically need support, which is – naturally – where SUSE makes its money. As well as direct commercial support, the company offers various consultancy services including SUSE Assist. SUSE Assist, the jewel in SUSE's support crown, offers on-site services from a certified professional for companies that can't afford to have a specialist on their staff.

As well as its Desktop and Server variants, SUSE Linux Enterprise comes with the option of add-on extensions, including those that make it suitable for use in point-of-sale environments and high-availability extensions where required along with real-time and thin-client variants.

Stability and support do come at cost to flexibility, however: compared to the community-driven OpenSUSE, SUSE Linux Enterprise has fewer packages available by default, removing many of the packages that aren't well-suited to a more professional environment.

The final tick in the box for SUSE is its SUSE Studio platform, which provides a means for users – Enterprise or otherwise – to customise Linux distributions and create everything from Live CD images to VirtualBox images and even Amazon Elastic Compute Cloud (EC2) instances tailored to their needs.

While SUSE is our pick at present, it continually trades places with Red Hat as the two attempt to out-do each other; before committing to one, be sure to check out what the other's offering too.



SUSE is a highly curated distro, tailored to its needs

ALTERNATIVE DISTROS

Red Hat Enterprise

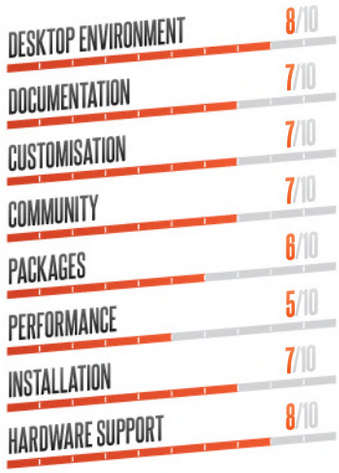
Like SUSE, Red Hat offers its Enterprise Linux variant in server, desktop and specialist variants, and boasts a healthy client list including ETH Zurich. Support is plentiful, and the company operates training facilities throughout the world.

OpenSUSE

OpenSUSE is the community-driven, fully-open variant of SUSE Linux. Sponsored by SUSE, OpenSUSE requires no support contracts or licensing and often provides newer features.

Ubuntu

Like SUSE and Red Hat, the company behind the software – Canonical – offers varying support contracts and training options, while its software compatibility is top-notch.



BEST FOSS

Puppet

An open-source configuration management utility designed to support heterogeneous networks of Unix-like and Windows machines, Puppet is a powerful automation tool for sysadmins of Enterprise-class infrastructures.

Chef

An alternative to Puppet, the Ruby- and Erlang-based Chef integrates well with commercial cloud environments including Amazon's EC2 and Google's Cloud Platform, and works as a local install for managing internal infrastructure.

Docker

Docker provides the ability to easily and quickly deploy applications inside isolated software containers on Linux. Compared to a traditional virtual machine, a Docker container has significantly lower overheads.

Lynis

Designed for those who take a proactive approach to security – but, it has to be said, a handy tool for the black-hat crowd as well – Lynis provides a means to audit Linux and other Unix-like systems for security vulnerabilities, and can also check for configuration errors.

SUSE Studio

While SUSE Studio is most commonly used by the SUSE and OpenSUSE teams, its ability to customise and deploy operating system images can be used with any Linux distribution and can make a system administrator's job significantly easier.

BEST DISTRO FOR...

Security

While the mainstream media worries its readers over black-hats, security-focused distributions are a vital tool for the good guys

Kali Linux

For years, BackTrack Linux was the king of Linux distributions for those doing security audits and penetration testing. In 2013, however, the project was forked into Kali Linux. Created and maintained by Mati Aharoni and Devon Kearns of Offensive Security, Kali is a ground-up rewrite of BackTrack and a worthy successor to it.

Based on Debian, rather than the Ubuntu origins of its predecessor, Kali includes pre-installed copies of the most popular security utilities, including network sniffer and analyser Wireshark, port-scanning tool nmap, password cracker John the Ripper and even the Aircrack-ng suite for penetration testing of wireless networks. Its repositories have plenty more choices available, in addition to more sedate applications and utilities.

Kali has everything you need for full security testing

Where Kali differentiates itself from the competition is in compatibility: as well as 32-bit and 64-bit x86 hosts, the team behind it have worked hard to bring Kali to the more popular ARM-based platforms out there. Builds are already available for devices as diverse as the Raspberry Pi and Samsung Chromebook, with more builds arriving on a regular basis. Considering the very low cost of some of these devices, Kali's support helps lower the barrier to entry considerably over distributions which require more expensive hardware to run.

Perhaps the most impressive compatibility feature, however, is a Kali sub-project dubbed NetHunter. Currently available exclusively for Google's Nexus Android smartphones and tablets, Kali NetHunter provides various wireless penetration testing tools usable directly from the device – a great tool for discreet testing without having to lug an all-too-obvious laptop around.

Combined with options to install to a hard drive as well as run from memory and an attractive desktop which lends itself well to every-day use, Kali is the obvious choice for anyone with an interest in network security – despite its reputation as a script-kiddie's play-thing, it's actually pretty tight on the security front.

ALTERNATIVE DISTROS

BackBox Linux

For those who don't need ARM support, the x86-only BackBox is a great alternative. Based on Ubuntu and featuring the lightweight Xfce window manager, BackBox is powerful yet attractive and comes with a selection of pre-installed utilities.

Wifislax

For penetration testing of both wired and wireless networks, the Slackware-based Wifislax is brilliant thanks to integration of many unofficial hardware drivers and firmwares not normally part of the mainline kernel.

REMnux

REMnux specialises in reverse-engineering of malicious software. Tools are provided for memory investigation and analysis of various executable formats as well as documents and even web content.

DESKTOP ENVIRONMENT	6/10
DOCUMENTATION	7/10
CUSTOMISATION	5/10
COMMUNITY	5/10
PACKAGES	9/10
PERFORMANCE	9/10
INSTALLATION	6/10
HARDWARE SUPPORT	9/10

BEST FOSS

Lynis

Created by Michael Boelen, the author of Rootkit Hunter (rkhunter), Lynis is a fully open security audit tool. As well as checking for vulnerabilities, Lynis has the ability to find misconfigurations with reports that can prove to be extremely useful when hardening a system.

nmap

A tool so famous it ended up with screen time in *The Matrix Reloaded*, Fyodor Vaskovich's (real name Gordon Lyon) nmap should have a place on every system. Its rapid network mapping is incredibly flexible and can be individually tailored.

OpenVAS

The Open Vulnerability Assessment System (OpenVAS) started life as a fork of Nessus under the name GNessUs. Now, it's one of the leading vulnerability scanning and management tools – and it's entirely free and open-source.

Wireshark

While Wireshark – formerly Ethereal – has its competitors in the packet-sniffing arena, its friendly user interface and powerful analysis and filtering tools are second to none. Wireshark is useful for general network diagnosis as well.

Metasploit

This framework is invaluable for penetration testers. When a scan has revealed a vulnerability, Metasploit can attempt to exploit said vulnerability; proving or disproving its existence quickly and easily.

BEST DISTRO FOR...

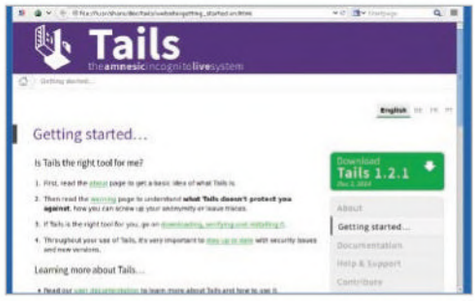
Privacy

Stay private and keep your information safe with these Linux distros specially built to put your mind at ease when working online

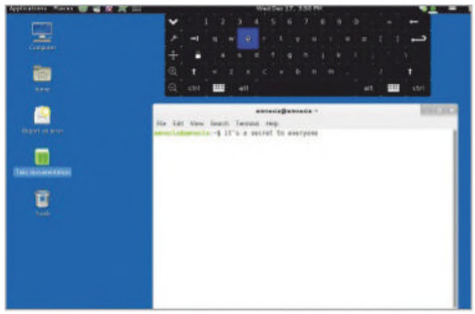
Tails

Privacy can be extremely important on the Internet, and it's only going to become more of a concern as time goes on. With more ways to leave your trace on the Internet, and more companies wanting your details to sell you ads, it can be tricky to remain truly anonymous. This can be essential for some people – whistleblowers, victims of stalking and people writing unpopular opinions on Twitter. It can also be handy for just buying a gift for your other half to avoid the inevitable targeted ads that an incognito mode won't stop.

Tails can help you with all of this, and makes it fairly easy to do in the process. It manages this through many careful layers of security and privacy considerations – firstly, the entire system runs in RAM and does not use any disk-bound swap partition. The RAM is then completely written over when Tails is shut down, leaving no trace of what you were doing or using.



Run Tails from a disc or memory stick to get the best use



All traces of user activity are removed as part of the shutdown process

All of its networking is run through Tor, so your IP is masked behind at least a dozen encrypted servers. Failing that, the default is the Tor browser, which also has the same software, meaning that whatever you're looking for, it won't get traced back to you. You can also use secure chat clients to keep your location safe, there's PGP email support built-in so you can send private mail and there's also just a full suite of normal programs like LibreOffice and GIMP, so you can use the distro in any other way.

You can install Tails, but it's designed to be live booted and that will guarantee maximum privacy at the same time. Give it a go today to find out just how easy it is to remain anonymous.

ALTERNATIVE DISTROS

iprediaOS

An alternative to Tails that relies more on the I2P network than Tor, yet still provides an environment where you can stay completely anonymous. It's also live, so you can test it out before committing to it.

Whonix

A very different private distro, anonymously connecting you online via an anonymous terminal and anonymous server, giving two levels of security to maximise privacy and stop any incoming attacks.

Liberte

A Gentoo-based Tails alternative that's not had much development recently but is still extremely private and very secure. It uses Tor and other software to keep the user safe from prying eyes.

DESKTOP ENVIRONMENT	8/10
DOCUMENTATION	7/10
CUSTOMISATION	6/10
COMMUNITY	10/10
PACKAGES	6/10
PERFORMANCE	10/10
INSTALLATION	4/10
HARDWARE SUPPORT	7/10

BEST FOSS

Tor

In the movie *Sneakers*, an intrepid group of Bay Area hackers bounce their signal off multiple servers and satellites to avoid detection. Tor is essentially this, sending your requests through several secure and encrypted servers.

Tor browser

Built using Firefox and Tor, using the Tor browser is an easy way to stay completely anonymous online without the need for booting into a private distro. It's so good its used by default in Tails to make sure you stay private.

ClawsMail

This is a PGP encryption for your email clients, including Thunderbird, that lets you send messages in confidence. It also works on its own, just in case you want to leave even less trace of its existence on your system.

KeePassX

Manage your passwords with KeePassX, the cross-platform password manager. It allows you to store a lot of data in a highly encrypted database that can only be accessed via your password – once it's accessed, you can even search it.

Florence Virtual Keyboard

A virtual keyboard that avoids any keylogging programs to make your computer just that little more secure. It can also be used if your keyboard is missing and broken, and is extensible and customisable.

BEST DISTRO FOR...

Rolling release

For those who don't like to be beholden to formal release schedules, rolling-release distributions promise to never get out of date

Gentoo

Named for the speedy Gentoo penguin, Gentoo and Arch have long been rivals. Both offer a true rolling-release development methodology, meaning that the latest updates are brought to the entire user base simultaneously – ensuring that no installation is ever out of date, and that installation need only occur once – and both feature a BSD-inspired ports-like software distribution platform.

For the Linux purist, the hands-off approach of Arch is likely to appeal, but for the average user Gentoo is a gentler introduction to the world of rolling releases. First released back in 2002, the distribution has a considerable fan base who appreciate the team's still-rare approach to development and software releases.

The other main advantage to running Gentoo is that its software is compiled from source directly on the user's system via the Portage manager. This means no waiting for package maintainers to build and upload a package for your platform, and that the software which gets installed can be optimised for your specific processor architecture – enabling performance boosts where generic compilation would drain power from the system. The trade-off, of course, is that compiling from source typically takes longer than simply installing pre-compiled binaries from a package archive.

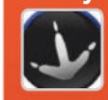
Like Arch, Gentoo's installation process has been tricky – although plenty of community help is available in documentation, IRC channels and mailing lists – but the relatively recent release of a Live USB variant makes it far easier to try. While its popularity has waned in recent years, Gentoo remains a great choice for anyone who wants a highly customisable system, and while it can be tricky to install it's a process that – in theory – should only ever have to happen once.

Support forums and an IRC channel are linked right from the desktop

Installation is slower but gives you better-optimised apps

ALTERNATIVE DISTROS

Sabayon

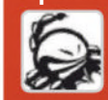


A Gentoo variant, Sabayon retains the rolling-release ethos but is a lot more welcoming. Designed to work out-the-box, Sabayon loses a little in flexibility compared to its upstream parent but is still a powerful distribution.

Arch

Arch can be unwelcoming to newcomers; first boot drops the user at a console session and installation is a process of following the instructions on the wiki. Once installed, though, it's lean, fast and extremely flexible.

Aptosid



Aptosid offers a familiar environment and rolling-release development. Taking Debian's unstable branch as its parent, Aptosid includes a custom kernel and retains compatibility with Debian's Free Software guidelines.

DESKTOP ENVIRONMENT	7/10
DOCUMENTATION	6/10
CUSTOMISATION	10/10
COMMUNITY	8/10
PACKAGES	9/10
PERFORMANCE	10/10
INSTALLATION	4/10
HARDWARE SUPPORT	9/10

BEST FOSS

GIMP

The open-source answer to PhotoShop, the GNU Image Manipulation Program has recently introduced a single-window mode to combat criticisms of its unfriendly userface, bringing its power to a new audience.

Audacity

Supporting multi-track mixing and with more filters and utilities than you could imagine using, Audacity helps prove that Linux is no slouch when it comes to creative work and that it can hold its head up there with the proprietary platforms.

Firefox

Although under fire for perceived bloat – ironic, considering the project was founded to deal with perceived bloat in the Netscape browser – the Firefox browser, now on version 34, remains a popular choice among users.

LibreOffice

Created following Sun Microsystems' acquisition of OpenOffice.org, The Document Foundation's LibreOffice is now the default in many distributions, offering features and compatibility to please even the biggest Microsoft Office fan.

VLC

The strength of VLC lies in its flexibility. As well as the ability to play almost any audio or video format, it supports streaming over the network and the ability to record from various sources – including capturing a live view of your desktop.

BEST DISTRO FOR...

Live distro

The best live Linux distros you can boot up from portable media without installation

Knoppix

Knoppix is still one of the premier live distros, although competition has become fierce with other distros popping up that add something different to the mix. Knoppix has remained popular thanks to some core design choices, while updating in other areas to keep with the times.

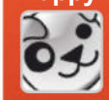
Knoppix positions itself as a showcase of everything that open source has to offer, and depending on what version of the distro you get, this can translate to having access to just about every known FOSS available on Linux without the need to install them – it has everything and the kitchen sink.

For having quite a lot of software, Knoppix boots and runs fairly fast. This is due to the way all the software is compressed and decompressed 'on the fly', allowing for 2 GB of the usual DVD to contain up to 9 GB of software that can be used at any time. Knoppix also has several custom boot options on a cheat sheet that will let you boot with different sound or display options, and even boot into the special ADRIANE interface for those who are visually impaired. Knoppix can be very handy to have installed onto a DVD or USB storage if you're regularly needing to quickly boot into Linux for some reason on various computers. It's not the best for sysadmin work, but it can do many other Linux-only computing tasks.

DESKTOP ENVIRONMENT	8/10
DOCUMENTATION	7/10
CUSTOMISATION	5/10
COMMUNITY	7/10
PACKAGES	10/10
PERFORMANCE	9/10
INSTALLATION	4/10
HARDWARE SUPPORT	9/10

ALTERNATIVE DISTROS

Puppy Linux



A very special and tiny distribution that, while very light for normal computers, is best suited for giving ancient PCs some usefulness. It's based on Ubuntu usually, with a quite custom kernel and a different set of packages.

WebConverger

WebConverger allows you to set up a dedicated web kiosk for something like an Internet cafe, running a modified version of Firefox.

Porteus

A very fast live distro that takes up only 300 MB of space, and is optimised to run from live media. You can add modules for extra software if needed.

BEST FOSS

Clonezilla

The best way to clone your hard drive, Clonezilla supports full hard drives as well as partitions and can then be used to restore disk images in the future. It can be used on its own but it's best to use the live version.

GParted

Format, edit, resize and basically do anything you want with your hard drive and partitions using GParted. It's included on most live CDs because it's excellent at doing this task and is also easy to use.

Wicd

An excellent and easy to use networking utility that can be used for both wireless and wired, it makes connecting to and managing networks easy, in the past we've had it win our network manager group test.

TestDisk

If your backups have failed or something else has gone horribly wrong, TestDisk can recover your data from a hard drive. It supports all major file system types and works from the terminal.

You can download Knoppix from www.knopper.net/knoppix

FASTER BETTER SERVERS

UP YOUR COMPUTING POWER WITH AN UPGRADED OR BRAND NEW SERVER THAT YOU CAN BUILD YOURSELF

While big business and big data may be utilising mainframes more of late, the concept of servers is not going away any time soon. Servers are an integral part of any system, however large your IT infrastructure is. Whether it's inside the data centre or tucked away in your (well-ventilated!) cupboard at home, there are still a lot of uses for servers in 2015 and 2016.

For the office you may want to save a bit of money and create something perfect for your needs that you know exactly how to maintain. For home you may just want to enhance your setup and make the entire network more efficient. For both it's a great way to separate certain aspects of your network to control it in a more efficient way.

There are many components of a server that you need to keep in mind, but it boils down to an appropriate hardware selection and a good distro for the task at hand. In this tutorial, we are going to concentrate on file and web servers, two base server systems that can be expanded and modified in multiple ways to best fit the situation you are in.

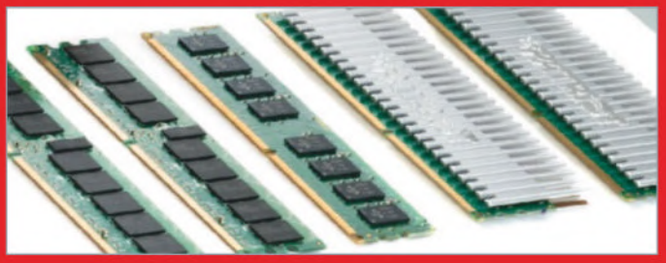
As we're teaching you how to build a better web server, we will first take a quick detour to tell you what you should know if you want to upgrade your current server so that it can compete with the new tech.

UPGRADING TO A BETTER SERVER

If you have a server, an upgrade may be all it needs to run better

You may already have a server, in which case instead of actually building a better server from scratch, you may want to just upgrade your server to be more efficiently than it was before. There are several ways of doing this depending on how you want to improve your server, and most of them require a hardware upgrade. If you decide to go down the hardware upgrade route, refer overleaf to see the kind of hardware that we recommend and learn some quick tips on how to install it if you're new to system building.

The easiest upgrade is storage space, especially for file servers. For Linux systems you can quite simply just add an extra hard drive into the case, as long as you have room in terms of spare SATA cables and power. Once installed, reboot your system and you can start adding the hard drive under /etc/fstab so that it automatically mounts to a specific location – in this case, the location on the filesystem which needs a bit more storage. Otherwise, you can create a clone of the system using Clonezilla (**clonezilla.org**) and then restore it to a larger hard drive with almost no change in the way it works.



Above RAM is much more important in a server than a desktop PC, as you need to serve several people

For other system hardware, you need to ask yourself which section is slow and perhaps needs upgrading. If it's a little slow for certain operations and computational tasks, your first port of call should be upgrading the CPU. Depending on how forward-thinking you were when building or buying the original system, the motherboard may support newer processors than the one inside it. Find out the socket information and start a search for a new CPU. While you'll need enough RAM to support the CPU and whatever the server is being used for, you'll always need more for one handling web services than file serving. You can easily replace these kind of parts without having to reinstall Linux.

If you're doing heavy computational tasks and can use hardware acceleration for it, look at getting a new video card to support it – although not many servers will even require one, let alone a good one.

If you've reached the limit of your current motherboard, it's time to gut the system and get a new mobo, CPU, RAM and GPU if you need it – backing up important files and settings is a good idea before you attempt this as Linux may not be able to work with completely new hardware without a reinstall.

Otherwise, if you need a software upgrade then refer to whatever guide is relevant to you in this feature on how to install and setup a new distro.

CHOOSING HARDWARE

What kind of hardware will you require to build a better server?

The hardware in a server is a very important consideration for building your system. Servers handle different requests to a normal desktop machine, often handling several people's requests at once. This means that the resource priorities have changed and these can even be different between various types of servers.

Software counts as well, of course, but without a decent hardware base, it will be tricky to have the server work as intended. Scalability and peak loads need to be considered as a future-proofing method, so always try and make sure that you have a bit more power than you need. With all that said, let's start looking at the individual components.

There are six main components you need to put thought into, and the four most important ones are the motherboard, the processor, RAM and power supply – the core components on any computer. As we mentioned, you need to think differently about what you need components-wise because resource usage is different.

A minor concern for some will be a graphics card of some kind, whether it's so you can directly interface with the system or do computational work that benefits from multiple different cores instead. You'll need a good storage solution for your build.

Motherboard

Motherboards for servers come in various styles. A lot of server boards will have two ports to connect a CPU to, which is good for servers used for small businesses or if you expect to get a lot of requests on a regular basis. These are more expensive than single-CPU systems, but the benefits in the long run for a big office server are more than worth it.

For home use, a single slot for a processor will do you fine for most cases, the main exception being a web server where you plan to have a lot of regular connections made to it. In this case, you want to keep an eye out for motherboards with plenty of storage and connection slots to make it as flexible and scalable as possible.

Above You won't need a GPU if your mobo has onboard graphics and you don't need multi-core processing

CPU

The most important thing for a server CPU is the number of cores – that's why dual-slots can be quite useful. More cores allows for more threads, essential if you plan to run VMs off a file server or several sites at the same time. Clock speed is not as important, but you should at least get one that is not ridiculously slow and comes with a decent cache.

With Intel's Hyper Threading, each core can work harder by creating multiple threads in each core. Conversely, AMD processors will offer more cores for a lower price, especially if you're on a budget.

RAM

A larger amount of RAM is more important on servers than it is on a desktop PC, enabling you to run more operations at once. Speed and latency is not so important, so gaming RAM with tweaked timings will not grant you a better system – in fact, it may be slightly worse since they don't have ECC. ECC fixes single-byte errors that make up the most common forms of data corruption in the RAM.

While ECC RAM can be important, it's more important in web servers and generally much more necessary in business and enterprise servers. On every level though, a larger amount of RAM is good.

PSU

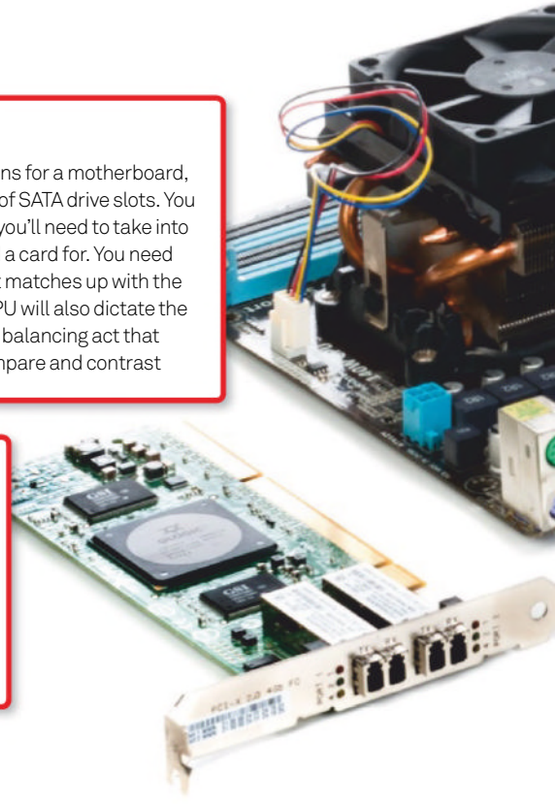
While it is best practice to never skimp on a power supply, it is near essential when it comes to server power. While you may need 1,000+ watts for your ridiculous 4K gaming rig (electricity bills be damned), you can be a little more reserved in the peak power for a home server, depending on its intended use. Look for power supplies with an '80 Plus' rating, as these ones have been through some level of certification to ensure that they have a degree of efficiency – this is a good idea for servers that are on all the time as they will save on electricity bills in the long run. Titanium and Platinum are the highest ratings, meaning they're at least 90 per cent efficient (95 per cent efficient for server power supplies).

MOTHERBOARD

When we talk about slots and connections for a motherboard, we're talking about PCI slots and plenty of SATA drive slots. You can add more SATA slots via a card, but you'll need to take into account anything else you'd want to add a card for. You need to make sure the motherboard's chipset matches up with the kind of CPU you want as well, and the CPU will also dictate the type of RAM you get. It's a multi-layered balancing act that may result in a sea of tabs while you compare and contrast

FIBRE CARD

Networking cards can be essential if your server is also acting like a more traditional network server, handling all your network data and even being used as a modem and firewall. There are plenty of different PCI cards for these kinds of tasks, including this fibre card for a bit more serious Internet use



Storage

Depending on your storage requirements, there are multiple solutions that you can use. At the very least we recommend you split up your storage with an SSD for the operating system and associated settings files, and use standard hard drives for storing everything else. This way, when the general files are not being accessed, the operating system can still run while drawing much less power.

Otherwise, your actual mass storage can be configured in multiple ways. You can have straight drives connected with JBOD for minimal complexity. Or you can start looking down the RAID route – mirroring in case of drive failure, striping to more efficiently use the space of two hard drives, or even going as far as RAID 5 and 6, which increases complexity but enables you to create one large, consistent storage space with redundancy failures. The more complex you go though, the more difficult it can be to maintain and the more catastrophic a major failure can be.

CASE

For server cases, you need to think of where you're using your server. At home, a silent case can be great, with padding to keep the noise down while it runs all the time. In the office, and assuming it will be taxed a bit more, you'll need to take into consideration proper cooling. Size is also a factor, as you need to fit your parts in

PSU

When picking a PSU you need to keep in mind a few things, such as what kind of connectors you need. This can depend on your motherboard, the amount of hard drives you're using, any extras like case fans and case I/O panels. If you want a better idea of what kind of wattage you will need, you can use this tool to figure it out: **bit.ly/1pjcjns**

RAM

You don't need RAM with heatsinks for server PCs, really – it's usually reserved for gaming RAM with tweaked timings. If you are concerned with the heat of your system and have a little more budget to spend, get RAM with some kind of heat dissipater

DVD DRIVE

They're more rarely needed these days, but it can sometimes be handy to have an optical drive for transferring data or to have ISOs installed to. If you're using your server to serve files and media, you can create redirects to be able to play DVD or Blu-ray movies, or even rip ones that don't come with any DRM at all

BUILD A FILE SERVER

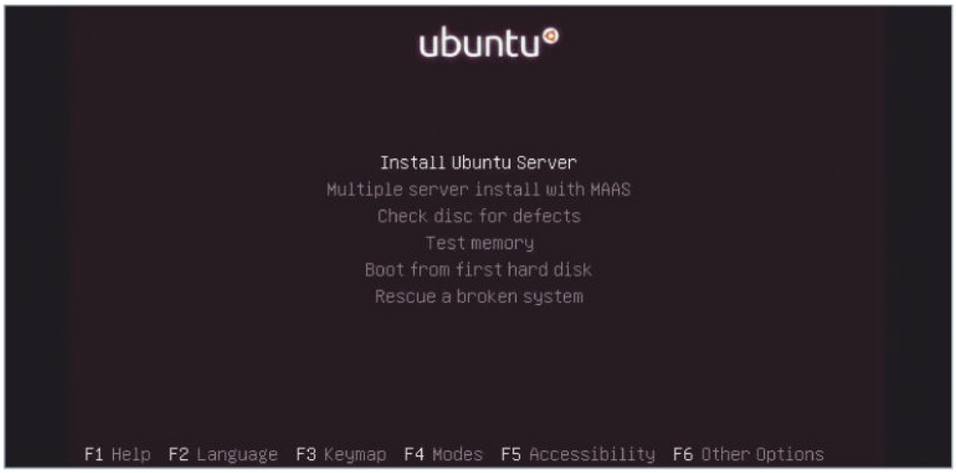
Store and serve files around your network or further

File servers are very useful for both home and business environments. For home, it's a good way to have a more low-power, dedicated solution to storing your media and backing up your systems, without needing to specifically turn on your desktop machine to get the files – a desktop machine that may use more power idling than a dedicated file server.

For enterprise, it can not only be useful for backups, but also provides for off-machine networked storage for individual users that can be accessed from within and outside the network. So, let's set a server up.

For a simple server type such as this, we're going to go ahead and use Ubuntu Server to set up the system. This means that if you have any experience with Linux, it should be easy to maintain and install more software on if you need to.

If you're doing the initial setup for a home server then installing it with a monitor attached will be much easier. Burn the ISO to an installable medium or boot it over the network if you have the facilities set up, then hit return on 'Install Ubuntu Server' to continue.



Installation

The installation for the server edition is different from the usual graphical installer of Ubuntu – it's a command line one, albeit with fairly straightforward options. After setting up your location, language and keyboard settings, it will try and detect your hardware for you. Give your server a name, set up your username and password, and then continue with the installation as directed.

Above You'll need to configure Samba in order to get shared folders working

Like the graphical Ubuntu, the server edition comes with options to automatically set up the partitions – by default, using the whole disc will create an install partition and a swap. If you want it to use a specific set of partitions, we recommend that you sort them out with GParted before trying to install them, and then assigning the partitions manually yourself.

During the installation process you will get some extra questions about whether you need a proxy or not; set that up as you wish and then it will ask about other services to install. As we're using this as a file server, make sure OpenSSH is installed so you can dial in from another machine on the network and ensure that a Samba server is installed, to make sharing files and such over the network easier and compatible with any Windows machines.

Finally, it will prompt you to install GRUB. Assuming this is a dedicated file server, you can let it overwrite the master boot record. Once that's done you will restart the system, so make sure you remove the live boot medium. After it loads up, you will be dumped into the command line to log in – as this is a server distro, there is no desktop environment.

First steps

Now you're into Ubuntu, we'll first get set up to SSH into the machine. For something like a home server it's best to set a static IP, and we can do that in /etc/network/interfaces. Open it up with:

```
$ sudo nano /etc/network/interfaces
```

... and change the primary network interface to be something like:

```
auto eth0
    iface eth0 inet static
        address [Desired IP]
        netmask 255.255.255.0
        gateway [Router address]
```

If you are using a wireless connection, make sure you switch it to wlan0 and then add in details for the SSID and password.

With the IP you've set, or using ifconfig to find out what the IP has been automatically set as, you can now SSH into your machine using the username and password that you set up. From a machine on the same network, type:

```
$ ssh [username]@[IP address of server]
```

Entering the password will grant you access to the same command line interface.

Shared folders

Now we can create a shared folder that the rest of the network can see and modify. First, let's create the folder that we want to use and put it in the normal home directory, with a usual:

```
$ mkdir ~/networkshare
```

EXTRA FILE SERVER USES

File servers can be useful for other things as well

Above left For a flexible, scalable server, consider trying MAAS: **maas.ubuntu.com/docs1.5**

Above right Ubuntu Server makes it easy to grab the software you need during installation

It's best if you don't use any spaces, to make the sharing simpler. Once done, you'll need to create a password for Samba. Do this by entering:

```
$ sudo smbpasswd -a [username]
```

It will ask you to enter and then confirm the password. Once that's done, and with the folder created, we can add it to the Samba server. Access the config file using:

```
$ sudo nano /etc/samba/smb.conf
```

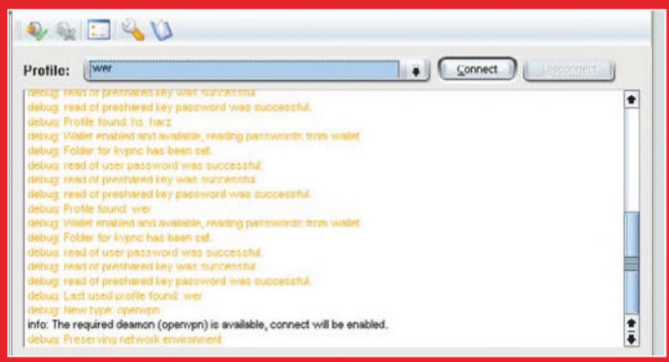
Go to the very end of the file and add something like the following to get the shared folder recognised by Samba:

```
[NetworkShare]
path = /home/[username]/networkshare
available = yes
valid users = [username]
read only = no
browseable = yes
public = yes
writable = yes
```

Save the file and exit, then restart Samba:

```
$ sudo service smbd restart
```

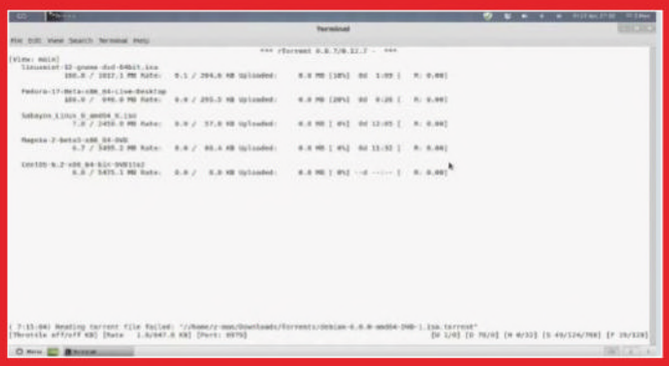
And finish by testing the setup with **testparm** to ensure everything runs okay.



VPN

Dialling in externally from a remote location to VPN into your server can have added benefits. Accessing your files remotely is one thing, but also being able to use a more unrestricted Internet service (yours) can be handy if you're stuck in a hotel or other location with strict browsing regulations.

Setting it up is not too difficult and requires the server to be connected to the Internet wherever it stays. The more users you allow to VPN from it, the more resources you'll require (including RAM and processing power).



Torrent server

With all the storage and possibly a connection to the Internet, you can set up the file server to also be a torrent server. This will enable you to give back to the community by seeding the latest distro torrents, as well as making sure you have the latest version of certain distros for you to install and test with.

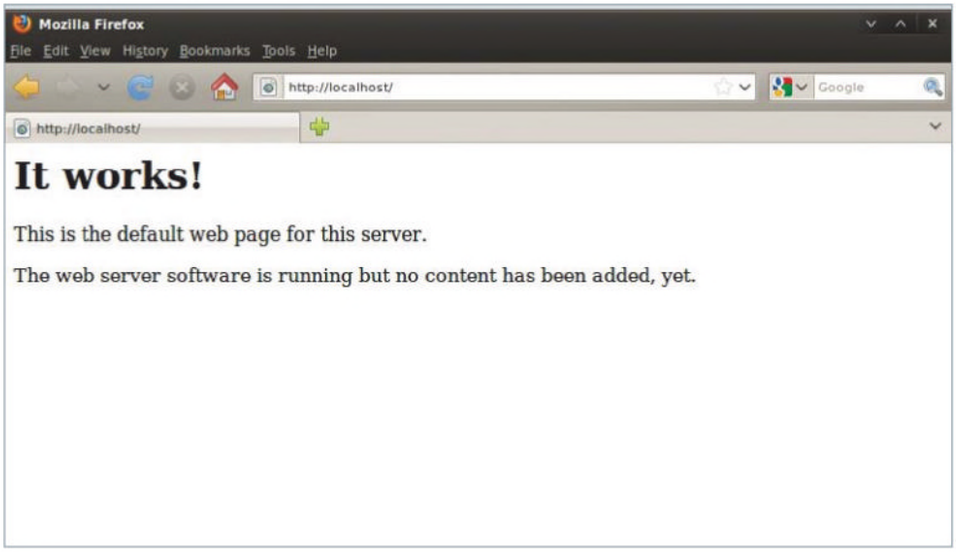
Just adding a torrent service will let you do this, and a good one for command lines is rTorrent. Not only can you view a useful command line interface with it, but you can also set a folder that it reads for new torrents.

BUILD A WEB SERVER

Host your webs services on a dedicated server that you control

Your own web server can be a useful addition to any system. If you don't have massive loads to worry about you can install it to your own custom-built server, or if you have a lot of scalable server space then you can build it on there with a very similar software setup.

We are going to use Ubuntu Server again for this, so follow our advice on the previous pages on how to get it set up and get to a point where we can start adding Apache services. Feasibly, you could have the server be both a file and web server in this way.



Above Give your web server a test once you've installed Apache

01 Install Apache

Once you've got your server set up and can SSH in, it's time to install the Apache web server . Do that using the following:

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

It will automatically set the domain address to the following: 127.0.0.1

02 Test server

If you have to install a GUI onto your server, you can test out that Apache is working by going to a browser in it and navigating to 127.0.0.1. Otherwise, from an external system, navigate to the IP address of your system in your browser and it should show the Apache confirmation page on-screen.

"With a web server you can now use it to host a website or to access storage from the server remotely over the Internet"

03 Install FTP

With a web server you can now use it to host a website or to access storage from the server remotely over the Internet. We can set up the latter using FTP, or in our case the secure VSFTP. Install it to the system using:

```
$ sudo apt-get install vsftpd
```

04 Configure FTP

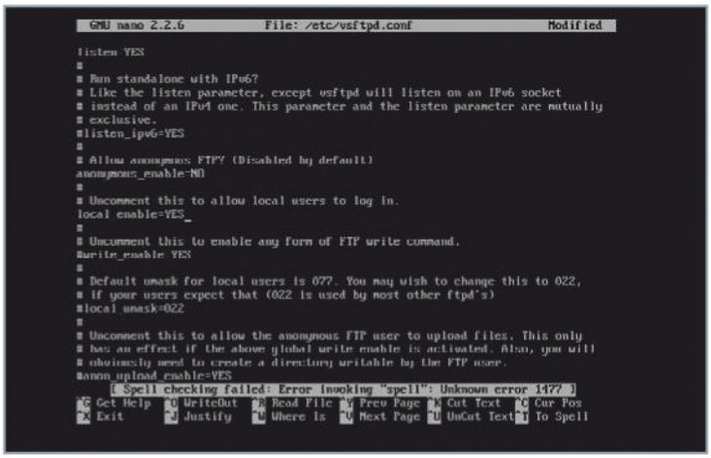
We can access the configuration file for FTP by using nano to open /etc/vsftpd.conf (**sudo nano /etc/vsftpd.conf**). From here we can configure it to match our uses, but first it is necessary that we increase the security just slightly before using it properly as an FTP server, just to be on the safe side.

05 Secure your FTP

The main change to make the FTP secure is to turn off anonymous users. In the config file, look for the line with 'anonymous_enable'. We want to change this to NO if it's not already there, just to make sure that there is a bit more security for the FTP server and that all of your content is kept private.

EXPAND YOUR WEB SERVER

Tailor your web server to suit all of your individual needs



06 **Local use**

While it will be great to access these files externally, we might as well have it so you can access the FTP internally too, just in case you prefer that to a shared folder. Find the line 'local_enable' and change it to YES to make sure it's accessible elsewhere.

07 **Edit files**

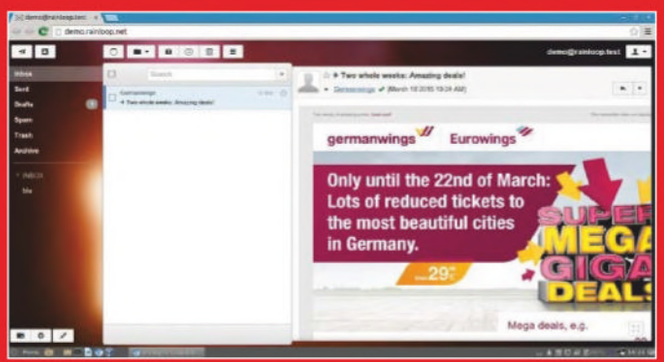
As we are only letting people we want onto the server, we can make it so anyone logged in has write access. To do this, go back into the config file, look for the line with 'write_enable' and change it to YES. You may also need to uncomment it by removing the hash.

08 **FTP Folder**

If you didn't create a shared folder for the previous server tutorial, now is a good time to create a dedicated directory for this. In the home folder, use mkdir to create a directory called whatever you wish for the FTP server to be able to access it.

09 **Restart server**

Once the config file has been fully edited and the folder is created, it is now time to start using it. Restart the FTP server using **sudo service vsftpd restart** and you will start to be able to access the folder. Any changes that you make to the configuration will require this restart to become active.



Mail server

Part of the benefits of this being a web server is that it enables you to also add your own mail server to it, or even host your own webmail client as well. Having Apache configured is the first step to this, and it is quite straightforward to then set up a mail server on top of that.

As for a webmail client, we recommend using RainLoop, which looks nice and modern and also lets you add other webmail services to it, along with your own webmail server.

Nginx server

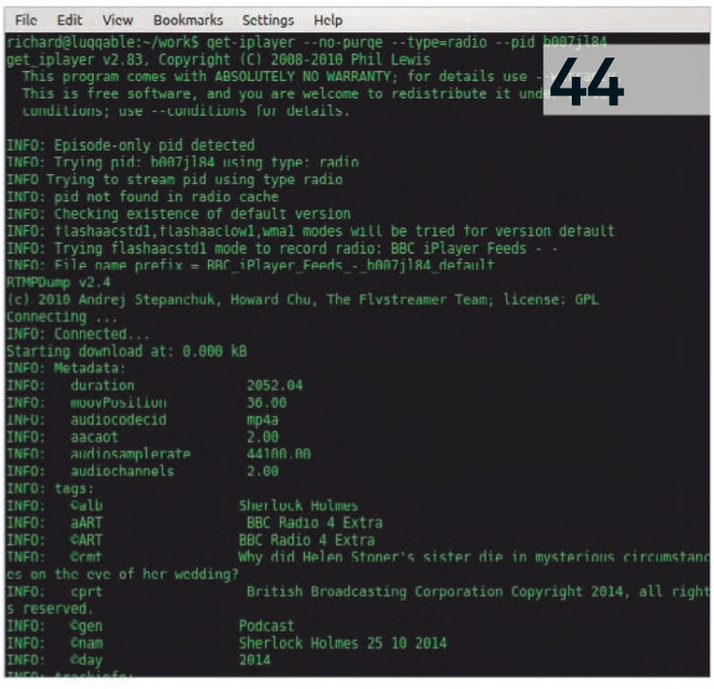
Instead of using Apache for the server, you can also look into and try out Nginx. Nginx handles processes slightly differently than Apache and can result in a lighter load on your web server. It's available in the repos of most distros, much like Apache is, so it can be installed in mostly the same way.

For a more complete guide to setting up an Nginx server, you can always refer back to one of our tutorials from Mihalis – it's inside issue 144, if you have it to hand, and on our website: **bit.ly/1AYKkx8**.

Tips & Tricks

Improve the way you work

"Photography has never been more popular, largely thanks to the availability of high fidelity DSLR cameras"

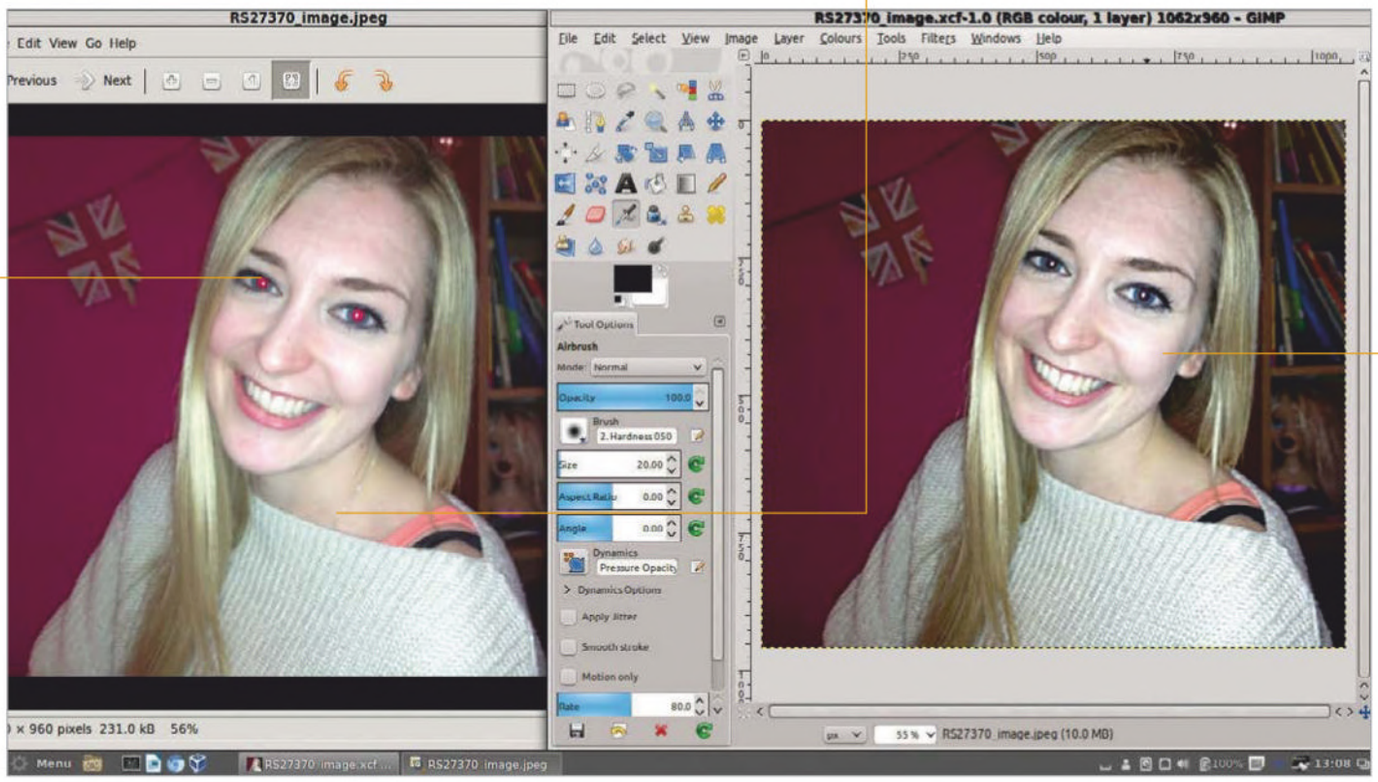


Page 60
Back up to the Cloud

Fix common photography issues such as red eye using in-built tools from GIMP

Learn how to remove minor blemishes and unessential items such as necklaces to improve overall picture quality

Smooth out and highlight skin using colour control and sharpness to make a photo more lifelike



Touch up photos using GIMP

Advisor

Rob Zwetsloot models complex systems and is a web developer proficient in Python, Django and PHP. He loves to experiment with computing

Resources

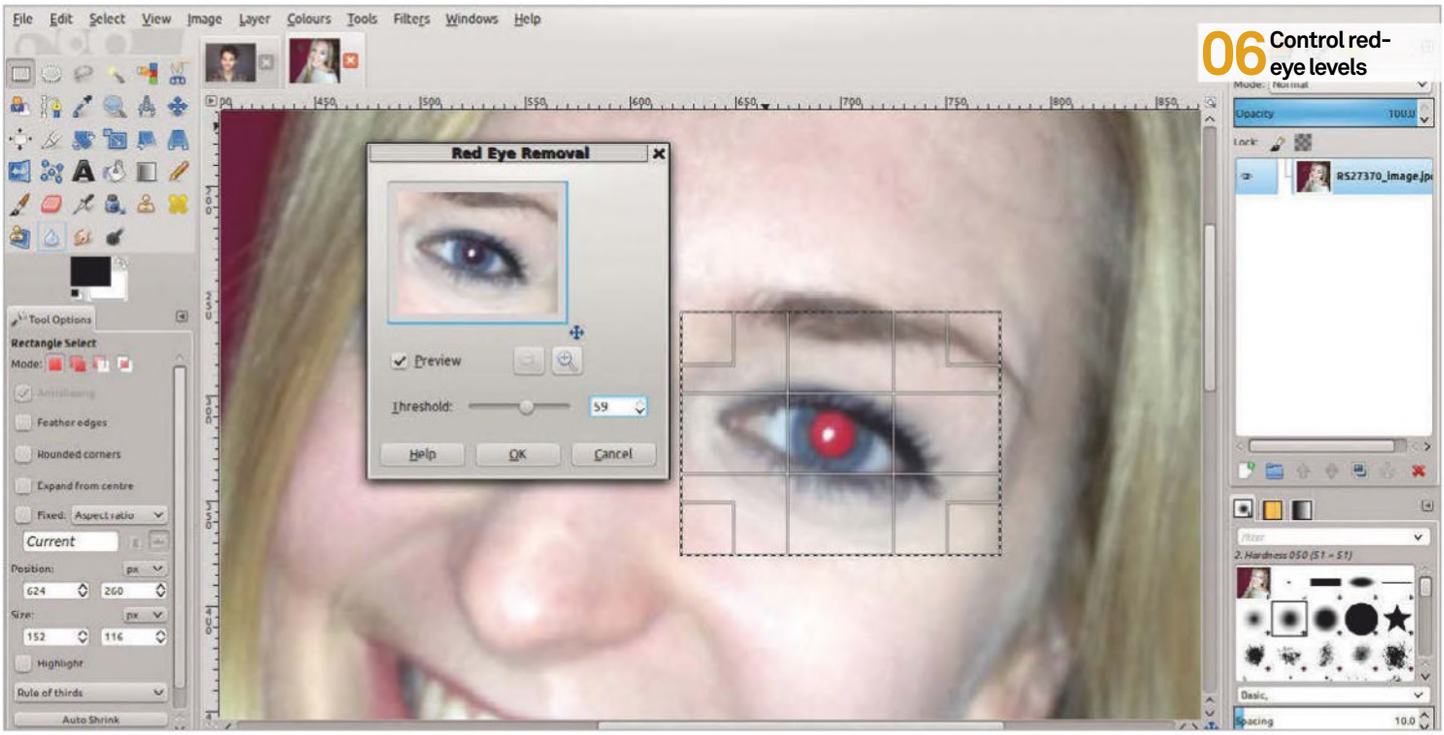
GIMP
gimp.org/downloads

Learn how to make professional photo enhancements with open source software

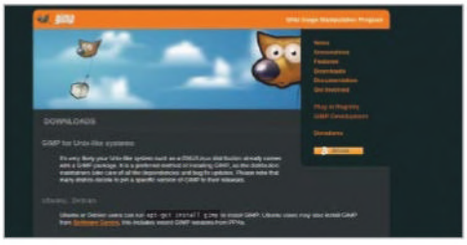
Hobby photography has never been more popular, largely thanks to the availability of high fidelity DSLR cameras and decent point-and-shoots, not to mention smartphones. Finally taking over from film cameras over the last five years, high quality digital photos are much easier to get off a camera than developing photos ever was.

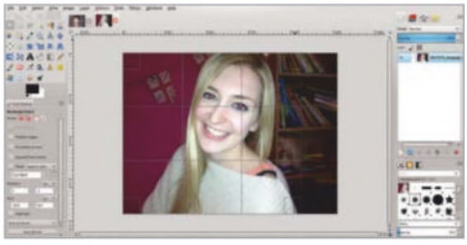
With digital photos also comes digital photo manipulation software, such as GIMP, which enables you to quickly perform professionally styled touch-ups and enhancements to photos in order to either really bring out the tones and lighting, fix any red-eye, control the colour temperature and more.

While Photoshop may be an extremely popular tool for photo editing, GIMP is definitely no slouch in that department. Having just about every feature you could get in Photoshop, with a few even being a bit better, it's most certainly enough for creating a great look with any of your photos.



06 Control red-eye levels





01 Install GIMP

GIMP is included on many Linux distros by default, so searching the Graphics category in the menus is your first step to finding it. Otherwise, it can be installed in your software manager just by looking for GIMP. If all else fails, head to **gimp.org/downloads** to get the installation files.

02 Work with RAW photos

Some DSLRs will allow you to work straight from the JPEGs, but others will also give you RAWs which can let you play around with the light levels and other fine camera aspects. GIMP can't edit these, so you may also need software such as UFRaw to properly import them.

03 Crop the image

Not essential for every image, but if you weren't shooting with a rule-of-thirds approach, you can always see if the image would look a bit better with one applied. Click on the rectangle select tool and set the Guides to Rule of Thirds.

04 Use the rule of thirds

The rule of thirds is used to position an image in such a way that certain aspects of a photo take up a third of the composition. This helps to make your photos look more dynamic and draws the eye to particular features. Play around to see what you want to highlight.

05 Remove red eyes

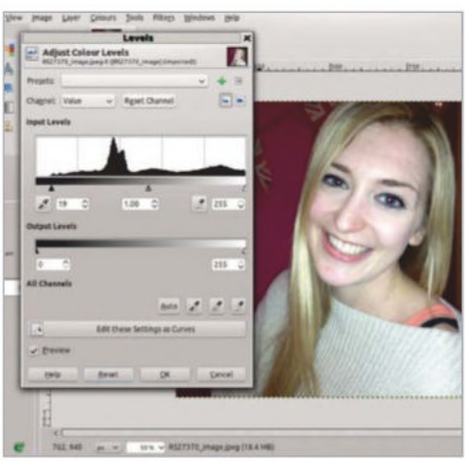
A common problem you may face with improper flash is red-eye. Luckily GIMP has a tool just for that. Use the Rectangle tool to select one eye at a time and then find the Red Eye tool in the enhanced sub-menu of Filters on the tool bar.

06 Control red-eye levels

Tweak the slider on the levels to remove as much red eye as possible without changing the whole picture. You can refocus the selection to be larger or smaller to try and get a better result. It can sometimes help to do both eyes at once.

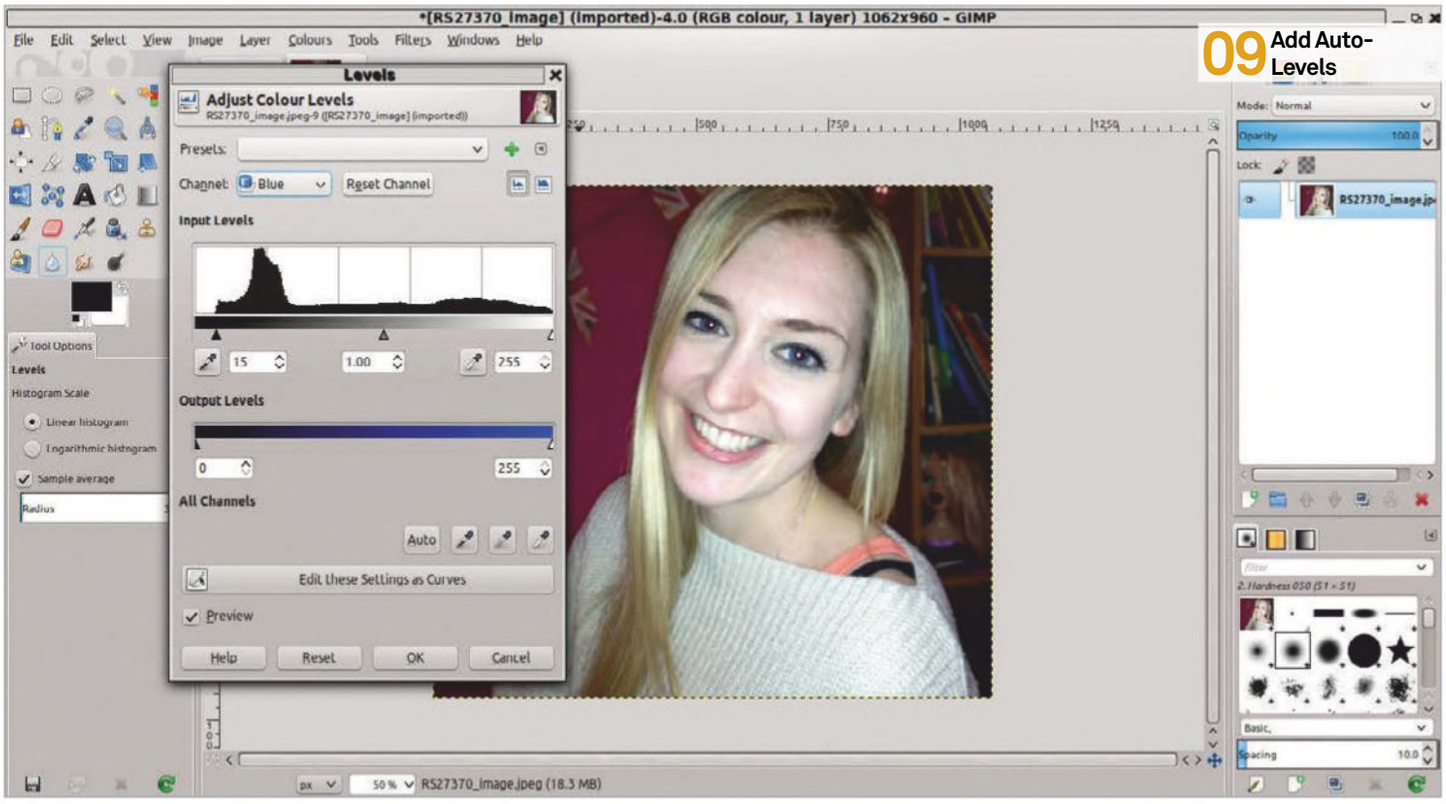
07 Vary highlights and shadows

Photos will have a range of colours at different levels, from the lighter highlights to the darker shadows. You can see a histogram and numbers for these settings by going to Colors>Levels on a 255 point scale; 0 is the darkest and 255 is the lightest.



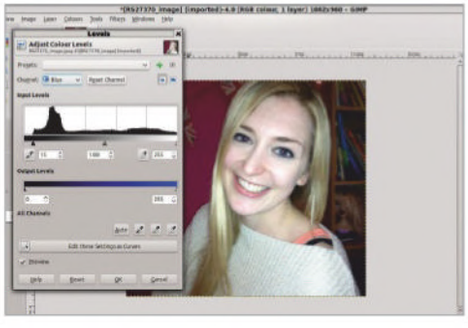
08 Adjust the overall range

Here you can see that the histogram doesn't cover the entire graph. A quick fix, and one that will instantly increase image quality, is to drag the shadow slider up until the beginning of the histogram. In this case it made the photo slightly darker but easier to pick out some detail.



09 Add Auto-Levels

There is an Auto-Levels tool that will automatically do any basic corrections on the photo for you. For some people and photos, this may be all you need to do to enhance your photo. Sometimes though, you might want to do a little more level editing to ensure maximum quality.



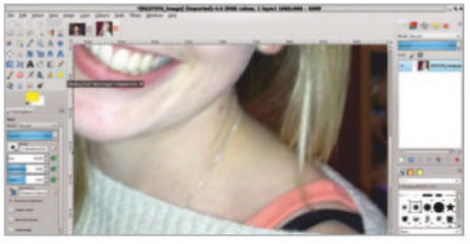
10 Alter the colour range

Back on the levels editor, find the channel selector. Here you can find the individual RGB levels. Editing these individually can create a slightly better tone profile across the picture. You can also edit the Blue or Red levels to make the image colder or warmer, or correct the white balance.

"Use the Eraser tool to remove the red areas from the skin"

11 Correct skin blemishes

There's a big debate going on right now over the beauty industry's use of airbrushing to make models look perfect. We're not really equipped to debate that in **Linux User & Developer**, but we feel it's fine to have a look at covering the odd skin blemish if you need to.

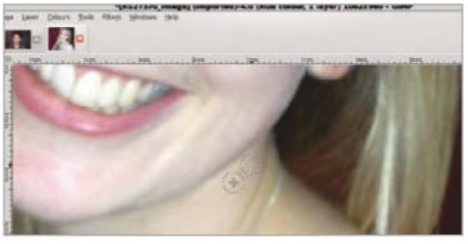


12 Use the healing tool

With skin blemishes like red patches, moles and freckles, you can easily cover them using the Healing tool. This takes one area of a photo and uses it to create a natural gradient. It's the plaster symbol on the tools.

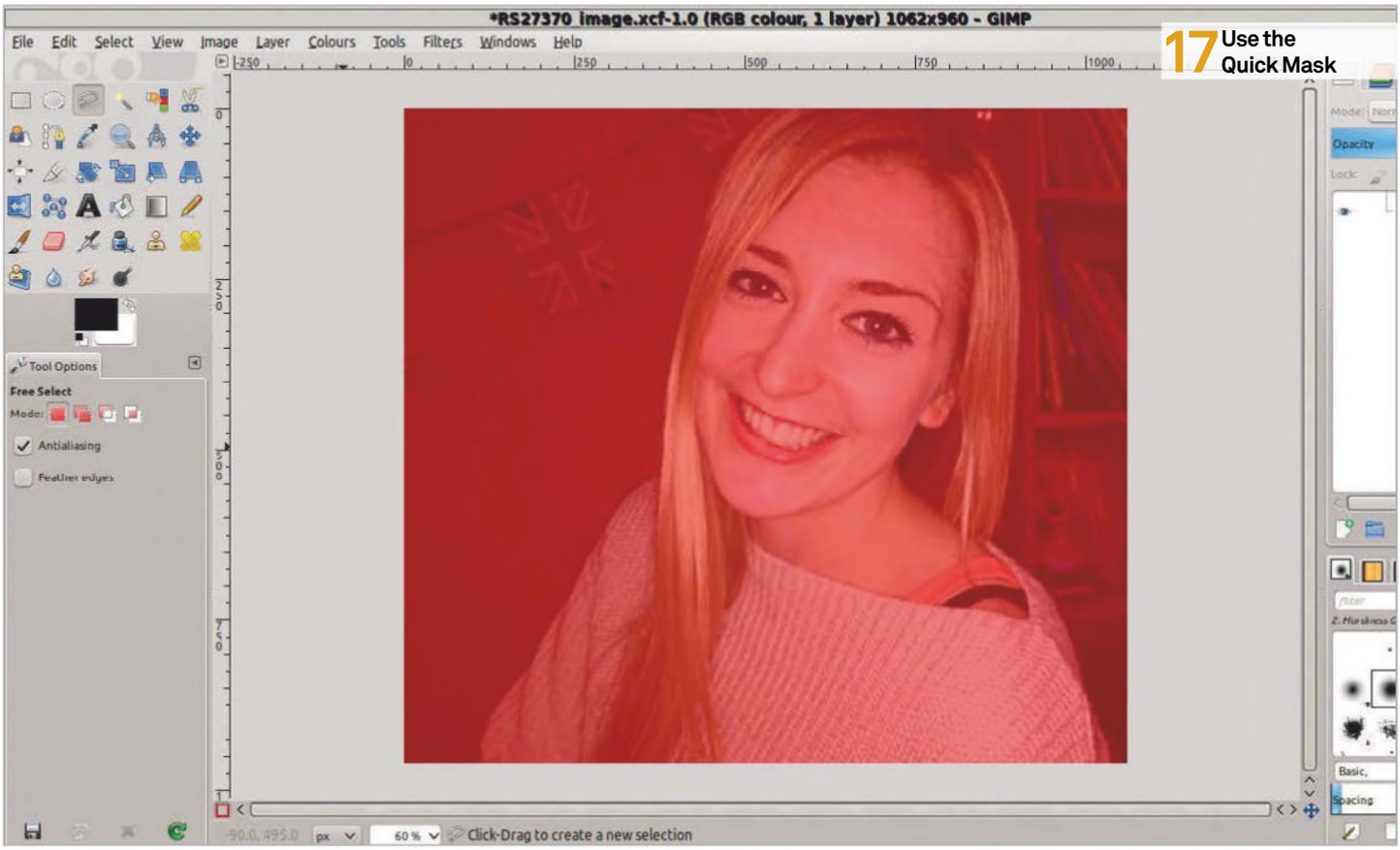
13 Set up the image

Grab the Healing tool and zoom in on the picture. Select a patch of skin next or very near the blemish – change the brush size if need be, depending on the size of the photo and blemish. Hold Ctrl before clicking. This selection will move with your painting to vary the healing tool's colours.

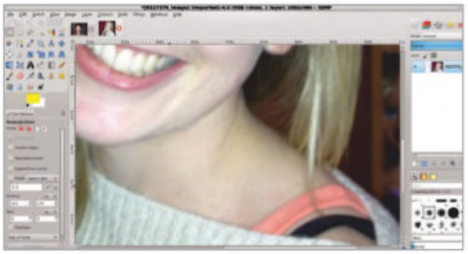


14 Heal blemishes

Paint over the blemish as you normally would any other colour using the paintbrush tool. You may need to reset the initial point of copy at points to avoid using the background or another part of the picture to cover up the target area.



17 Use the Quick Mask



15 Clean up the photo

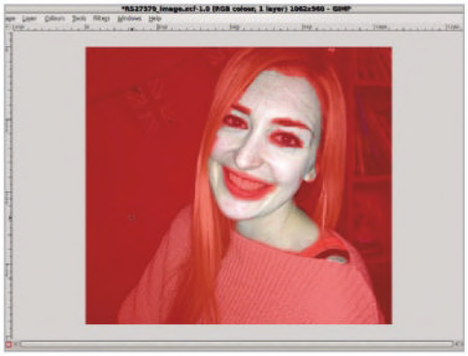
The Heal tool can do a lot more than just remove a mole. In this example we've removed the necklace from our model's neck. On a larger scale it can be used to clean up the background of a photo a bit more naturally than cutting out or guessing colour profiles. It's not a perfect tool as it can only estimate, but it's still very smart and gets better all the time.

16 Enhance the details

The Unsharp Mask works very well on photos with small details or where the makeup is key. Go to Filters>Enhance to select the tool, which will automatically bring out some of the details in the photo. Use small values and experiment; it only needs to be subtle.

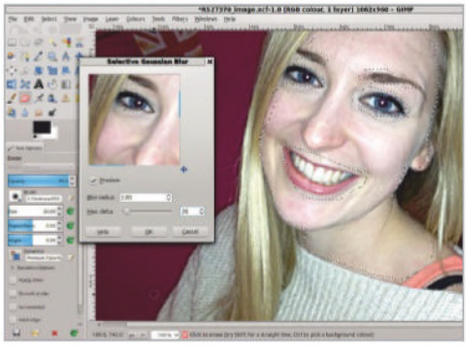
17 Use the Quick Mask

So we've removed some basic blemishes, but you can also smooth out the skin a little using the Quick Mask tool. Go to Select and then Toggle Quick Mask to cover the image in a layer of red. Don't panic, we'll be removing the red hue once we are done.



18 Do some preparation

Use the Eraser tool to remove the red areas from the skin, avoiding the hair, eyes and mouth. After that's done you can turn off the mask and it will create a selection of what you deleted, without the Joker-esque look that occurred when the mask was on.



19 Grab Gaussian Blur

The final step to smooth out the skin is to use the Selective Gaussian Blur in Filters>Blur. Play about with the levels again but try and keep them small. Go too far and you can make the skin slightly resemble plastic.

20 Beautiful photos

With a bit of practice and some creative uses of these tools and some others, you can really make any picture look much better than the original, without overdoing it and giving the model a completely different appearance.

Learn to code BASIC with FUZE

Discover the language that started it all with FUZE, the Pi-powered programming and electronics platform

Advisor

Rob Zwetsloot models complex systems and is a web developer proficient in Python, Django and PHP. He loves to experiment with computing

Resources

FUZE BASIC kit fuze.co.uk

A lot of the great British coders from Generation X are usually considered to have got their start thanks to the BBC Micro and the BBC Computer Literacy Project in the Eighties.

At the time, the cheap hardware was affordable for schools and allowed kids to code in 'Beginner's All-purpose Symbolic Instruction Code', or BASIC. This simple language opened up programming to kids of all ages in schools and created a British computing boom. Times have slightly changed and after a bit of a struggle, computing is back into British schools. While they won't be learning BASIC anymore, the language still exists and is an excellent way to teach programming. The FUZE line of hardware aims to bring that back with special Raspberry Pi-powered machines.

Not only can you code in the BASIC language but you can also work on creating physical projects using an excellent custom IO board attached to the FUZE device.

We'll show you how to use BASIC for programming and how to create a traffic light system.

01 Hook up FUZE BASIC
Some of the FUZE models already have a Raspberry Pi built into it, so all you need to do is hook it up to a monitor and power source to get it going. Make sure the SD card is placed into the slot before you plug it in using the supplied power adaptor.

02 BASIC interpreter
The FUZE will boot into a desktop of a modified Raspbian. Here you can do your normal Raspberry Pi-related activities, but what we're interested in is the FUZE BASIC option on the desktop that takes us to the BASIC interpreter.

03 Say hello
Our first foray into a programming language needs to be done right with a proper "Hello World" statement. It's very simple to do in BASIC and just requires you to write:

```
>PRINT "Hello World"
```

04 Look around you
While this version of BASIC has been updated since the BBC Micro days, you can still perform some of its more simple tasks and the infamous GOTO 10 command. You can do this by executing commands like this:

```
>10 PRINT "Hello World"
>20 GOTO 10
```

05 The modern way
Press Esc to end the loop. BASIC has seen some improvements in this version, so we can create this endless loop of Hello World by using a better loop statement in the code editor. To access the editor, press F2.

06 Modern loop code
The code for this is quite simple: we create a CYCLE, place the command within it and then tell the code to return to the beginning of the cycle with REPEAT at the end. In our case:

```
CYCLE
PRINT "Hello World"
REPEAT
```

07 Save and run the code
Once you've typed it in, you can press F3 to bring up the option to save the code. For now, just call it helloloop and press Return. Once that's done the code will run, looping Hello World over and over. Press Esc twice to end the loop.

08 Make variables
Creating a variable in BASIC is easy and is

"FUZE will boot into a desktop of a modified Raspbian. Here you can do all your normal Pi activities"

like most other languages:

```
A = 5
```

Use PRINT "A" to confirm it's worked. You can modify the value by adding a number or other predefined variables like so:

```
A = A + 5
```

```
A = A + B
```

09 Something more complicated.

Automatically increasing the value of a variable is easy. Let's create a fixed loop that prints out A as it increases from 1 to 5:

```
CLS
A = 0
FOR p = 1 to 5 CYCLE
      A = A + 1
      PRINT A
REPEAT
END
```

10 Wire up a project

The IO board offers a lot of extra functionality that's perfect for learning physical programming. While you can connect directly to the Pi's GPIO ports via the header, you can also use the custom ports. The tray at the top of the FUZE is perfect to slot the breadboard into, so do that now.

11 Wire up a light

Choose one of the LEDs from the pack of electronics – any colour is fine – and then insert the short end into one of the two pins along the top and insert the long end into the middle section. Grab a 100Ω resistor (brown black brown gold) and insert one end into one of the pins that are on the same vertical row as the long end of the LED.

12 Connect the light

Place the other end of the resistor on a pin in a different vertical column and then use a jumper wire to connect GPIO 0 to this end of the resistor on the vertical. Finally, attach a wire to GND on supply and put it on the same row as the short end of the LED.

13 Easy operation

To activate the light, exit the editor using Esc and type the following line to let it know to set GPIO 0 as an output:

"You can connect directly to the Pi's GPIO ports via the header"


```
>PinMode (0,1)
```

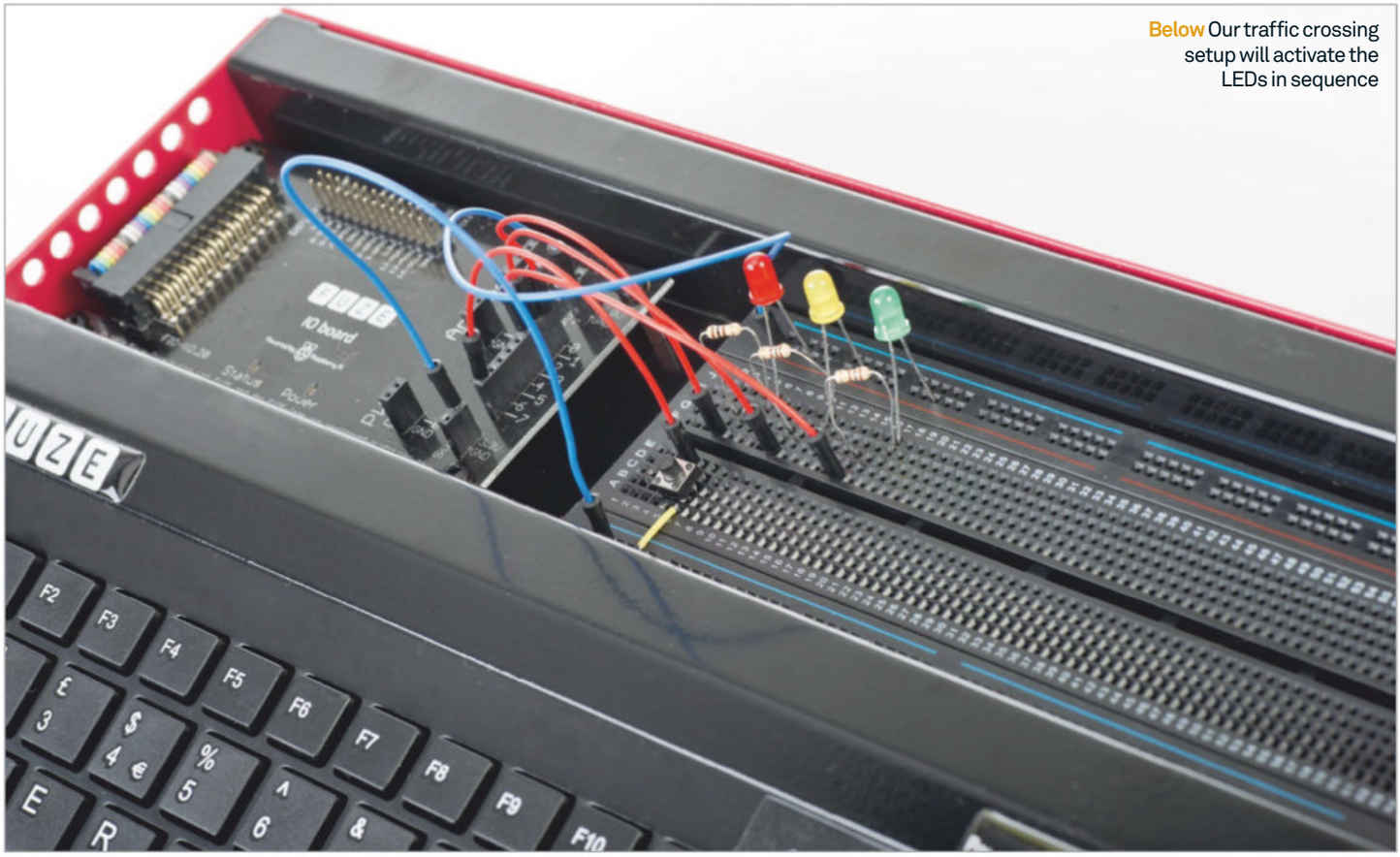
To activate it and then deactivate it, use:

```
>DigitalWrite (0,1)
>DigitalWrite (0,0)
```

14 Light coding

Now that we've got our light working, let's create a little code that will turn it on and off repeatedly:

```
CLS
PinMode (0,1)
CYCLE
DigitalWrite (0,1)
Wait (1)
DigitalWrite (0,0)
```



Below Our traffic crossing setup will activate the LEDs in sequence

```
Wait (1)
REPEAT
END
```

15 Bigger project

Let's ramp up the lighting setup on the breadboard and make it into a traffic crossing setup. Add two extra lights in the same way we added the original LED and attach them to GPIO pins 2 and 3. Wire up a button by attaching one end to pin 7 and the other end to a new 3.3V rail taken from the second set of power pins. The button will make the lights go 'red' in sequence, then after a pause go back to 'green' – see **Fig. 01** for the full code.

16 CLS

The CLS command clears the current display. This is helpful to make sure any errors or printouts from your code will show up without possibly blending in with outputs from previous programs or runs.

17 Use PinMode

The numbers on the PinMode variable handles the GPIO port (X) as well as whether it's an output (1) or input (0). It is therefore constructed like so for GPIO X as an output:

```
PinMode (X,1)
```

DigitalWrite is constructed similarly for output pins, with 1 being on and 0 being off.

18 IF THEN

When you create an IF statement you need to make sure THEN is appended to let BASIC know that the following code is for a True situation.

19 digitalRead

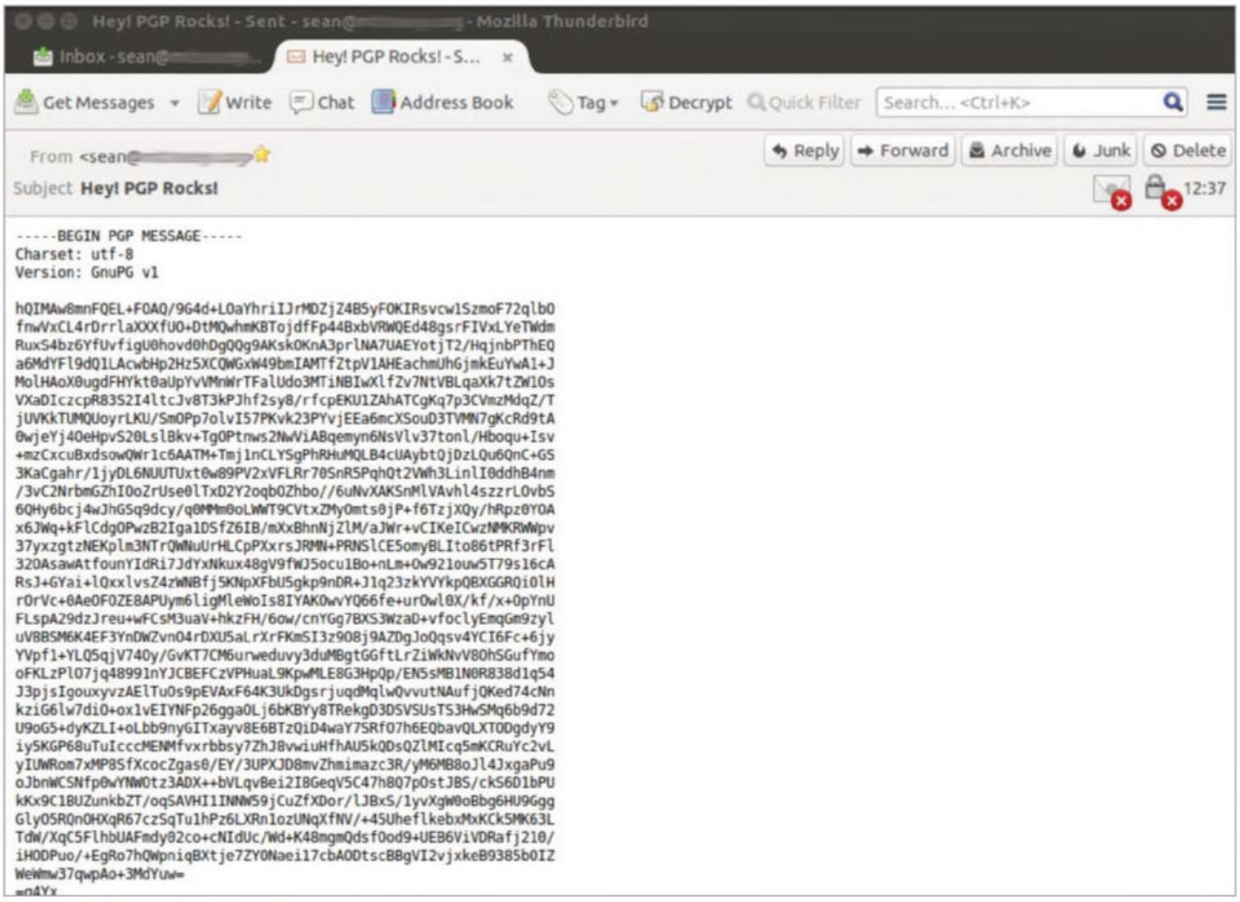
As pin 7 in this code is set as an input, we want to read when it's activated. Our button completes the circuit when pressed so the read needs to be 1.

20 ENDIF

Let the code know when the IF statement ends by adding ENDIF. This can make it slightly easier and clearer for figuring out where an IF begins and ends compared to other languages.

Fig 01

```
CLS
PinMode (0,1)
PinMode (1,1)
PinMode (2,1)
PinMode (7,0)
CYCLE
    DigitalWrite (2,1)
        IF digitalRead (7) =
1 THEN
            Wait (1)
            DigitalWrite (2,0)
            DigitalWrite (1,1)
            Wait (1)
            DigitalWrite (1,0)
            DigitalWrite (0,1)
            Wait (5)
            DigitalWrite (1,1)
            Wait (2)
            DigitalWrite (0,0)
            DigitalWrite (1,0)
            DigitalWrite (2,1)
        ENDIF
REPEAT
END
```



Encrypt your email with Thunderbird and PGP

Advisor

Sean M Tracey is a creative technologist at a leading digital agency. He spends a lot of his time living inside of Node.js, Python and Arduino

Resources

Ubuntu 14.04 / Debian 7.7

GPG www.gnupg.org

Mozilla Thunderbird mzl.la/1pimzQ4

Securing emails really isn't that difficult – with open source tools at our disposal, world-class encryption is at our fingertips

For years, we have been confident in our ability to transmit personal or sensitive information over the Internet securely. We bank, we shop, we send endearing messages to our sweethearts. It's understandable then that we want to feel secure in the handling of services that involve dealing with money, lifestyles and friends. SSL, SSH, HTTPS – these are just a few of the protocols that we use every day that try to assure us of their absolute security. Regrettably, recent revelations have revealed that these trusted protocols are not as secure – or rather, not as unimpeded – as we once thought they were.

In this tutorial we are going to look at how quickly we can use open source tools, in this case GNU Privacy Guard, to once again secure our private communications and ensure that the only people reading our messages are the people that we sent the messages to.

01 What is PGP? How does it work?

Sending messages securely has been attempted for centuries. Secrecy has been of paramount importance in times of war, matters of finance or great personal deeds. Despite its importance, there are always caveats to sending messages secretly – how can we be sure of the message's security? Are we certain that only the person who we intended the message for has actually received it and decrypted it? How do you encrypt something that can only be decrypted by the people you have selected?

Phil Zimmerman had a crack at answering these questions back in the early Nineties and PGP was his solution. PGP is a little different to other security methods; rather than having a single password or key that can encrypt and decrypt a file or message, PGP has two keys. These two keys are called the public and private keys. The public key is used to encrypt information and can be shared with anybody. The private key is one that you, as the creator of both of these keys, keep to yourself. When combined with the public key, the private key can decrypt the information that was encrypted with the public key.

This system is perfect for messaging because anybody can have anybody else's public key and encrypt data, but only the person that created the public key in the first place has the means to actually decrypt the encrypted information.

02 Get GPG

Encryption is hard on the best of days. Like many complex things in computing, we aren't just writing code to handle a task – we need to be certain that the maths behind the concept is sound, otherwise it's all for naught. Fortunately, today is one of those days where we don't have to worry about any of this because we have GPG (GNU Privacy Guard).

If you're running a modern Linux distro, you should already have GPG installed on your system, but let's assume that you don't already have GPG installed:

```
sudo apt-get install gnupg
```

This will get you up and running on an Ubuntu or Debian system. If you're using another system, you can download and build the source from **www.gnupg.org/download/index.html**.

To check that your installation is valid, type **gpg --help** into your terminal. If you see a list of options, we're ready to crack on.

03 Create our keys and identity

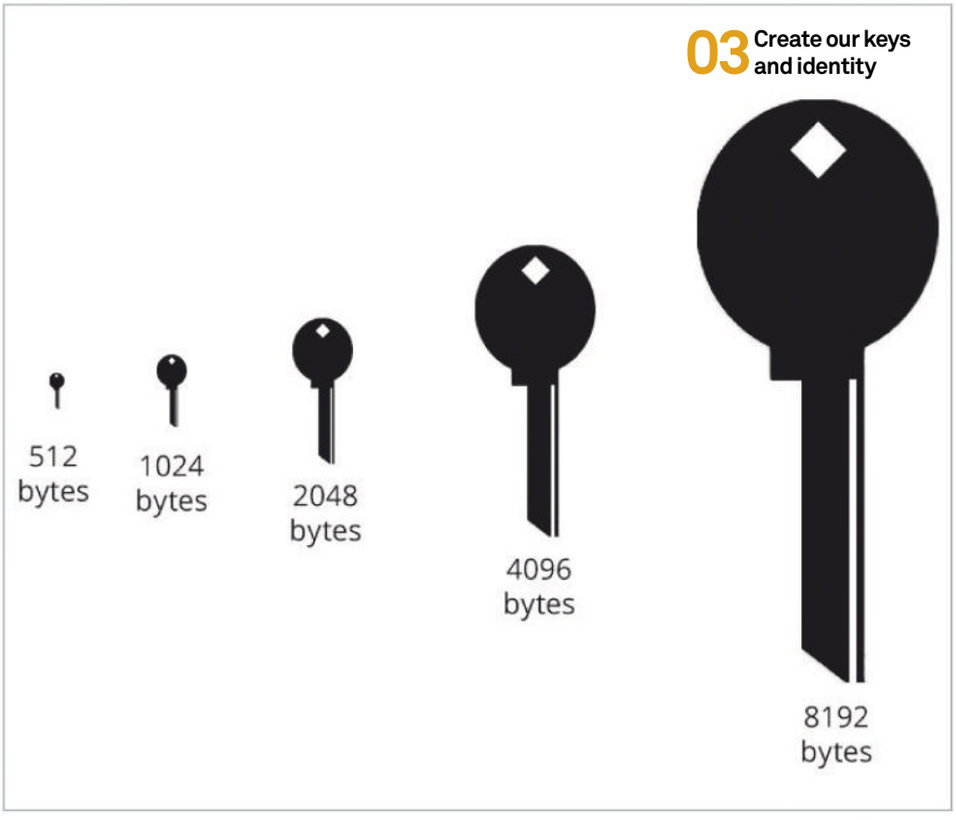
Now we're at the point where we can create a keypair – these are the public and private keys that were mentioned before. In your terminal, enter:

```
gpg --gen-key
```

...and you'll see the following options present themselves to you:

```
Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
```

We're going to be using option one, the RSA keys. You'll be asked what keysize you want for your keys. The sizes vary depending on your system, but always go for the upper bound of the options presented. This is because the larger the key, the harder it will be to brute-force the encryption in the future.



"This is because the larger the key, the harder it will be to brute-force encryption"

Next comes the expiration date for the key. We'll be creating a revocation key in a little bit, so an expiry date for our keypair isn't entirely necessary. If you don't like the idea of a key that can't expire, however, then enter an expiration time that suits you, otherwise just enter 0. GPG will then ask you whether you are sure you want to create a key that never expires; respond with yes.

As this is the first time we've used GPG, we don't have an identity to sign our keys with yet, so a prompt will ask us to do that now.

You'll need to enter your real name, a comment (often used to describe who you are) as well as a real email address. Once you've confirmed that you're happy with what you've entered, you will be asked to enter a passphrase. This can be anything – make sure that you jot down a note of it somewhere, as you will use it to access and allow access to your keychain later.

04 Entropy and randomness

Now our system will ask us to continue using our system while it generates random bytes. As we all know, a computer program that claims to be inherently random often isn't. When observed over time, patterns begin to emerge in the randomness. By using our system as GPG creates random bytes, a certain amount of entropy is created – true randomness. Our interactions with the system and the response to those outputs will help GPG to create a truly random key.

This may take a while, so just go about your business as normal. Play a game, make a cup of tea, watch that video of an adorable new puppy on Youtube and check back once in a while. (Tip: compiling something really helps speed the process along.)

After some time, you'll get something that looks like Fig. 01. This means that we've just successfully created our first keypair!

05 Revoke our key

Our key doesn't have an expiration date. This means that if, through some nefarious means, somebody manages to get access to our private key they would be able to read all of our encrypted data and there would be little that we could do to stop it from happening.

By creating a revocation certificate we can kill that problem. It is pretty easy to do this – just enter the following into your terminal:

```
gpg --gen-revoke [[EMAIL_ADDRESS_YOU_
ENTERED_WHEN_CREATING_KEYS]]
```

You'll be asked for a reason for generating a revocation certificate. We don't have a reason right now other than being overly cautious, but you can select any of the options you see fit and comment accordingly when asked.

Next you'll be asked for the passphrase you used when creating your keys, enter it and you'll see an output such as this:

```
Please move it to a medium which you can
hide away; if Mallory gets access to this
certificate he can use it to make your
key unusable. It is smart to print this
certificate and store it away, just in case
your media become unreadable.  But have
some caution: The print system of your
machine might store the data and make it
available to others!

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1
Comment: A revocation certificate should
follow

[[DATA THAT MAKES UP YOUR REVOKATION KEY]]

-----END PGP PUBLIC KEY BLOCK-----
```

Just as the output says, put that certificate somewhere safe – preferably not on the same machine and definitely not on the Internet.

06 Set up Thunderbird with Enigmail

So, we have a way of encrypting our emails. Painless, eh? But there's no way of sending or receiving them yet. Well, let's fix that now. We're going to install Enigmail for Thunderbird. Enigmail is a nifty, free add-on that will handle signing, encrypting, sending, receiving and decrypting all of our secure emails.

Just open Thunderbird, then go to Tools> Addons and search for and install Enigmail.

Restart Thunderbird, and once the restart has completed you'll notice that Engimail is now an option in your main application bar. Click on Enigmail>Setup Wizard and we'll have the keys that we generated moments ago now assigned to our email inbox.

To set up our keys for Thunderbird properly, you should answer the wizard questions like so:

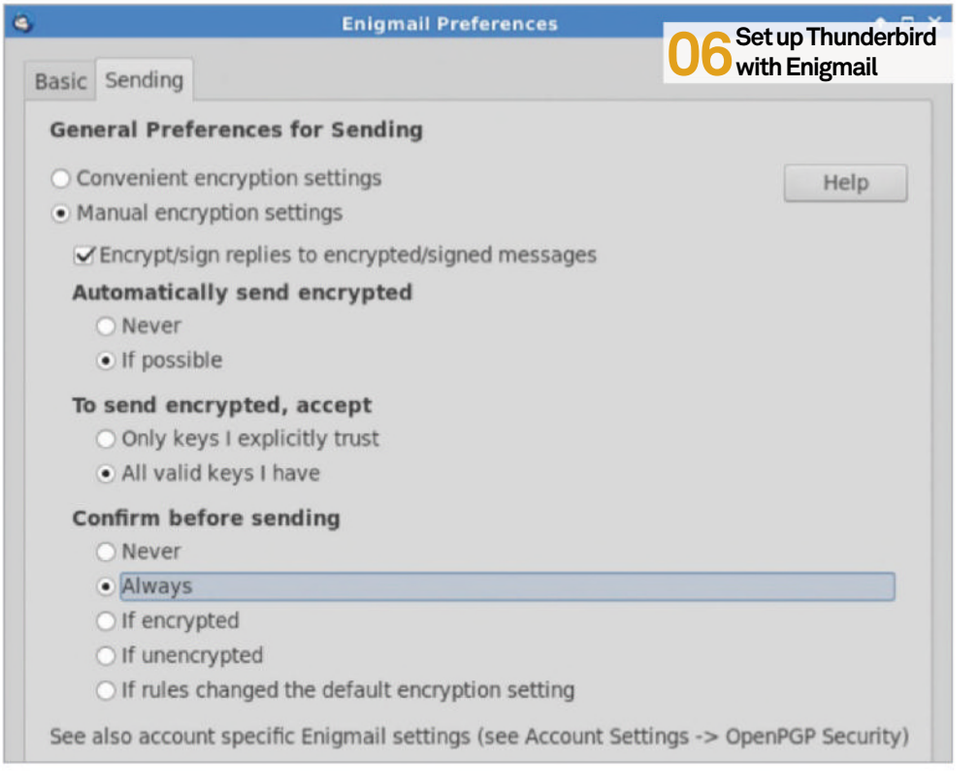
```
1 ) Which general mode do you prefer to
encrypt outgoing mail?

    - Convenient Auto Encryption
    and
    - Sign all of my messages by Default

2) Do you want to change a few default
settings to help Enigmail run better on
your machine?
    - Yes

3) I want to select one of the keys below
for signing and encrypting my email
    - [[ SELECT THE KEY WE JUST CREATED
    WITH GPG ]]
```

That's it then, we're all set up to send and receive encrypted email. At least, our mail client is. In order for people to be able to send us encrypted messages, they need access to our public key – otherwise how will we know what to decrypt if we don't tell them how to encrypt their message? It may seem counterintuitive to give away our own key when talking about security, but that's exactly what we want to do. In fact the more people that have our public key, the better – so let's share it!

Fig 01

```
gpg: /home/seanmtracey/.gnupg/trustdb.gpg: trustdb created
gpg: key 49A3764A marked as ultimately trusted
public and secret key created and signed.

gpg: /home/seanmtracey/.gnupg/trustdb.gpg: trustdb created
gpg: key 49A3764A marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   4096R/49A3764A 2014-11-12
      Key fingerprint = 84A5 21D7 2DB6 5BD3 8DB3  782D 82EF 6281 49A3 764A
uid                  Sean M. Tracey (Creative Technologist) <sean@XXXXXXX.org>
sub   4096R/042FE14E 2014-11-12
```

“ Interactions with the system and the response to outputs helps GPG create a truly random key”



06 Set up Thunderbird with Enigmail

usage. Assuming everything is behaving with your Thunderbird setup (and that you've opened the email with the key in Thunderbird), double-click on the .asc attachment. Thunderbird should recognise it as a public key and ask you whether you'd like to import it or not. If you answer yes, then you're set to go. However, if you're more of a command line person, then we can use GPG to add the certificate to our keychain as well.

Move the .asc file to somewhere accessible on your system (for example, your home directory) and cd to there with:

```
cd ~/
gpg --import someone_elses_key.asc
```

This will import the public key to your keychain, and now you're ready to send a message.

09 Give encrypted email a whirl

With Thunderbird, compose a new message just as you normally would. Make sure the person you send the email to is somebody whose public key is in your keychain. Just before you hit that lovely send button, tap the Enigmail button in the UI and check that this email will in fact be encrypted. If that's not the case, just select the options from the drop-down menu. If Enigmail recognises the recipient you intend to send the email to, it will automatically select that person's public key to encrypt the email with. If the recipient is not recognised, a prompt will appear asking you to select the appropriate key as you try to send the email.

Once the recipient has been selected and the message encrypted, the email will be sent on its merry way.

10 Receive emails

The neat thing about Enigmail is that when somebody sends you an encrypted message, it decrypts it for you on the fly! Same user experience, much better privacy.

11 Round up

We've looked at how PGP works, creating and sharing our own keys, setting up our email clients to handle those keys. and sending and receiving emails.

Despite the prevalence of other options, email is still king when it comes to digital communication, and still the vast amount of it is unencrypted. When asked about email security, people often answer it's not a concern because they've got 'nothing to hide'. Well, that's not the point – privacy is not a privilege, it is a right, and it is one that people should opt for more often.

"What we need instead is a central repository specifically for keys"

07 Share our public key

One way to get our public key out into the world is to email it to all of our contacts that we would expect to email us. If you only intend to send secure emails to a select set of people, this solution is probably best for you – but that would be no good to somebody who has never emailed us before. We could put our keys on a server, but how will people know how to find it? What we need instead is some sort of central repository specifically for keys. That's exactly what a public keyserver is for. You can add your public key to any keyserver you like – GPG comes with a preassigned default, **keys.gnupg.net**, and it's pretty well known so we'll upload our key there. All you have to do is enter:

```
gpg --list-keys
```

You will now get an output of all the keys you've created. You should see something along the line of "pub 4096/[[UID]] 2014-30-10". Copy the unique key ID you see and then enter:

```
gpg -send-keys [[UID]]
```

You'll then get:

```
gpg: sending key [[UID]] to hkp server keys.gnupg.net
```

This indicates that everything has gone well.

If you want to upload your key to a different server, point the same command to a different location:

```
gpg --send-keys --keyserver the.keyserver.address YOUR_KEY_UID
```

All that's left to do is test our setup, so let's send an email. First we need to pick a person to send it to.

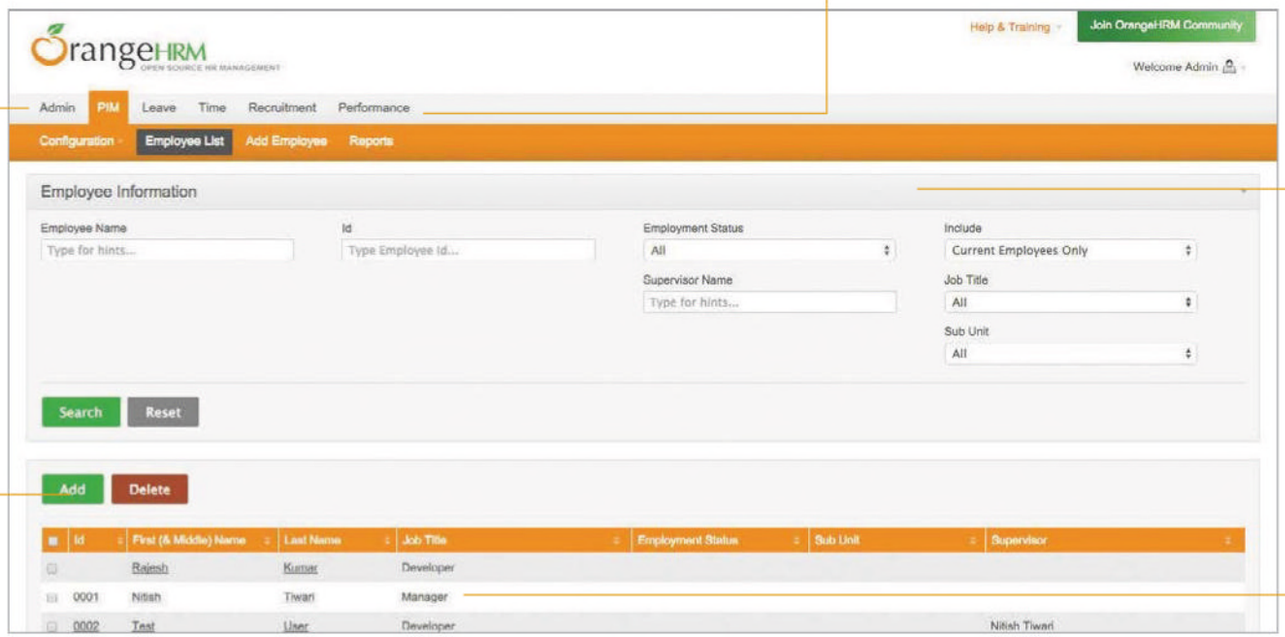
08 Add somebody else's public key

As mentioned before, we need a public key to send an encrypted email that only the intended recipient can decrypt. If somebody has sent you their public key through an email (which will have a .asc extension), there are two ways you can add it to your keychain for

The main modules of OrangeHRM. Clicking on one of them takes you to the respective module's landing page

Sub modules appear in this orange menu bar once you have selected one of the main modules

Search for an employee by name, ID or any other details. The search is very flexible allowing a search in spite of populated fields



Employee details are added here. The Add button reveals a new form which allows you to type in a new employee's details

List of employees already in the system. Clicking a name will reveal information about the employee

Simplify HR management with OrangeHRM

Employees are the most important part of any organisation and management tools are essential in maintaining efficiency

Advisor

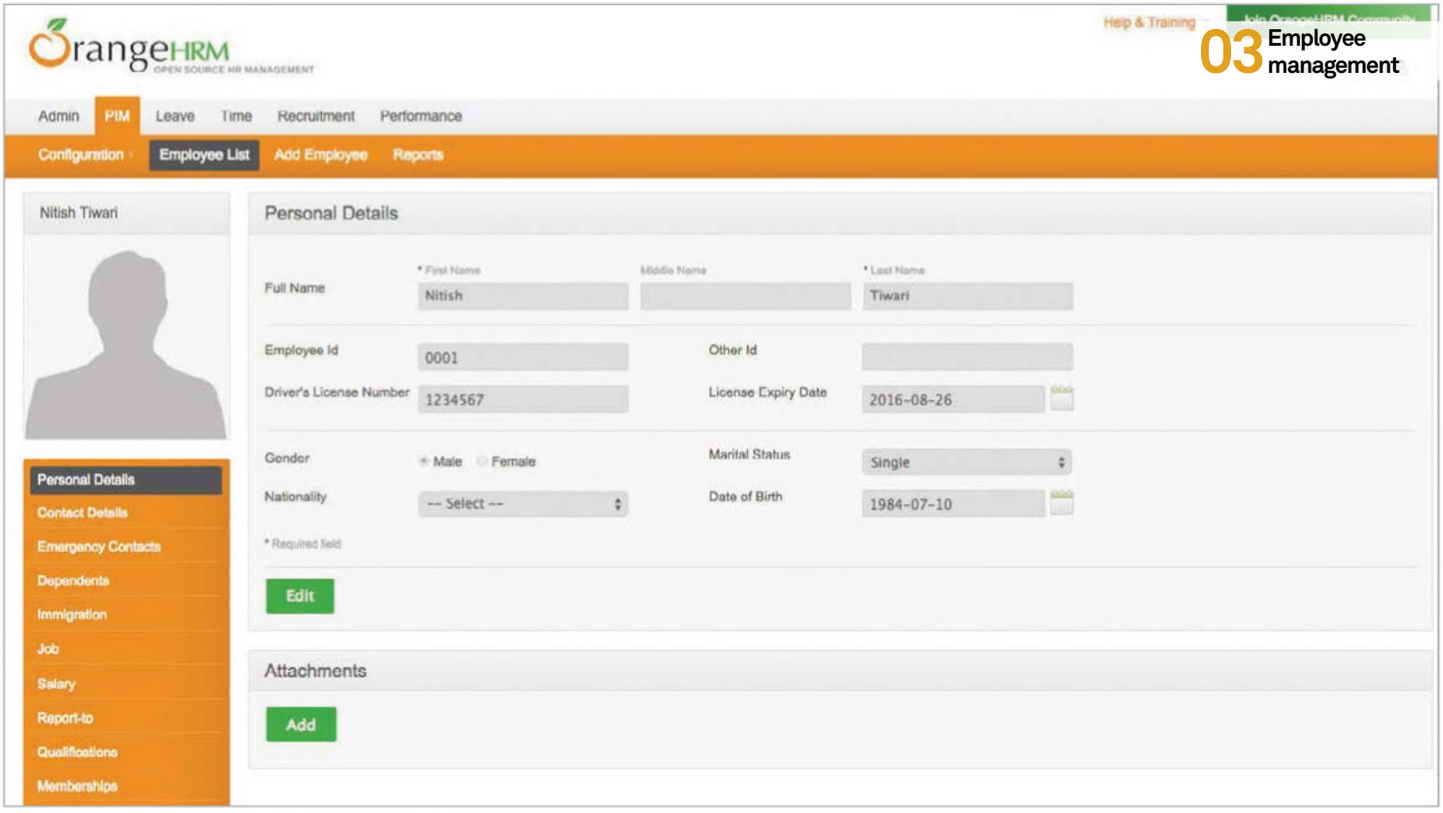
Nitish Tiwari is a software developer by profession and an open source enthusiast by heart. As well as writing for a leading open source magazines, he helps firms set up and use open source software for their business needs.

Resources

OrangeHRM www.orangehrm.com

For any organisation, whether a small one with few employees or a multinational corporation with several branches worldwide, managing human resources is always an important but difficult task. It is important because the employer needs to track key metrics and strategise accordingly to keep the employees in good spirits. It's also difficult because HR management is a diverse field with so many things to be managed; leaves of absences, performance, logged hours, employee profiles, salaries and a lot more. While organisations are now increasingly becoming aware of employee needs, leaving no stone unturned in making sure employees remain happy, the hunt for a great HRM tool sometimes proves to be the difficult part.

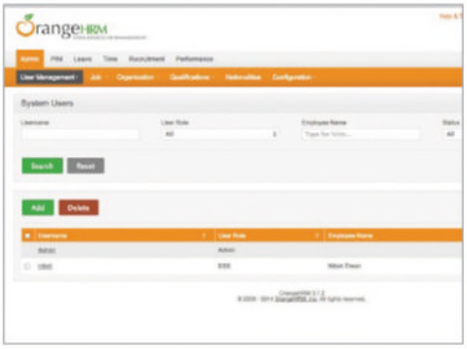
In this tutorial we will have a look at one of the most renowned and popular open source HR management tools – OrangeHRM. With the first beta release in 2006, it has continuously grown and is now used by one million users worldwide. OrangeHRM supports all the important aspects of HR management and is ridiculously easy to deploy and use. Given the ease of installation, configuration and use – and robustness – it is useful for all types of organisations, from startups to multinationals. In this tutorial we have used the stable release version 3.1.2.



03 Employee management

01 Installation

For Linux (and OS X) the source code is available for download as a ZIP file. Once you download it, just unzip and place the contents in the root folder of your webserver. You will need webserver and database preinstalled though – a LAMP/MAMP server, for example. If you also need to set up OrangeHRM on a Windows system, a bundled installer is available – complete with web server and database. Once you've downloaded and unpacked everything that you need, access the folder via a web browser – the URL should be something like: **http://localhost/orangehrm**. If everything is fine, the set-up wizard welcomes you with an option to freshly install OrangeHRM or upgrade existing set up. As you proceed for fresh installation, you will need to provide the database root access (if the database for OrangeHRM is not created) or just the details of database already created for OrangeHRM. You can then create the admin user ID and password. Finally, you have the optional registration before the installation finishes.

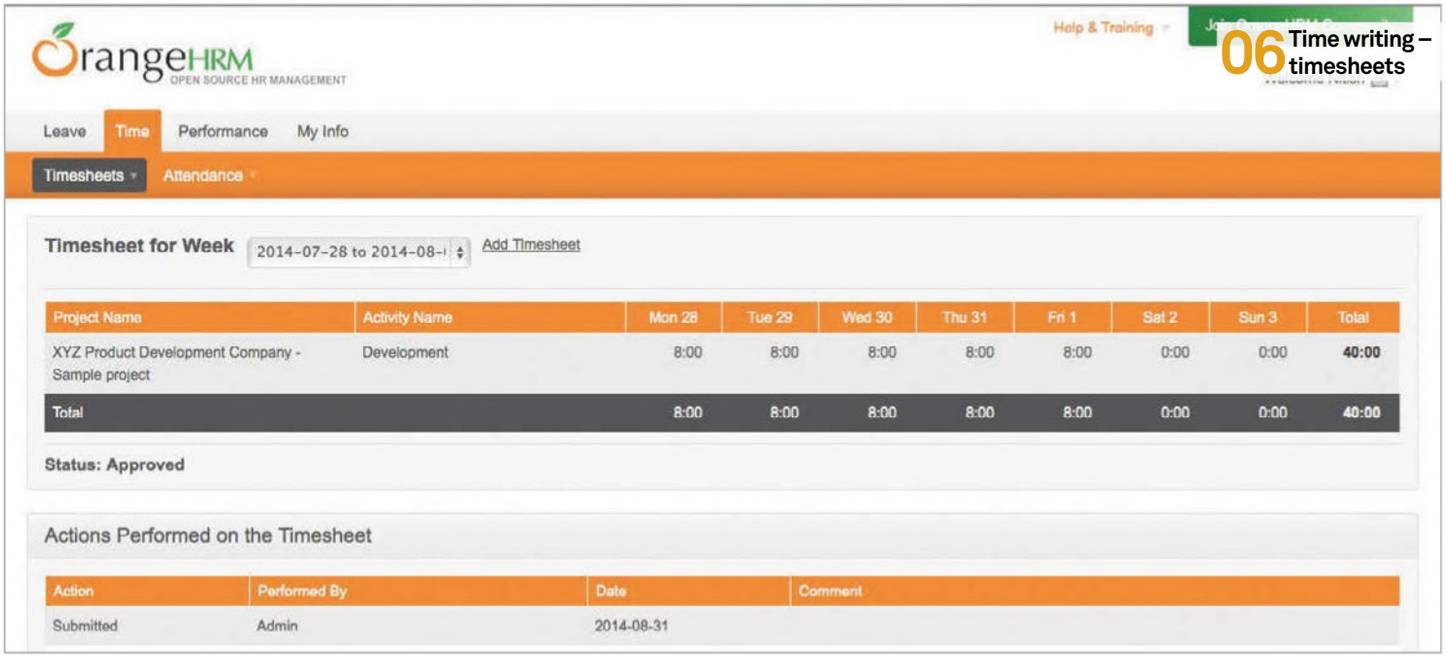


02 Administration and configuration

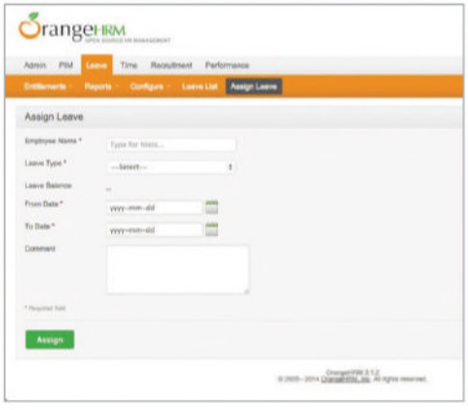
As you log in as an admin you can see several tabs corresponding to different functional areas. Under the Admin tab, you have the User Management, Job, Organisation, Qualifications, Nationalities and Configuration sections. You can set shift hours under the Job section. Employee qualifications can be set under Qualifications. Configuration lets you enable/disable different modules, configure email using sendmail or SMTP, and subscribe users to email notifications. While the other sections names are self-explanatory, there are few important tips you will find useful – User Management corresponds to the system users, and so you can't directly add a user. You will need to add the employee first (more about that in the next step) and they can then be added as a system user under the User Management section. A user can have only two roles: ESS (employee self service) and admin. Roles are not to be confused with job titles; there can be several job titles (which can be created under the job section).

03 Employee management

The PIM, or Personal Information Management, section is the place where you can manage the employees' data. Click on the Add Employee link and just fill the relevant details. If you select Create Login Details, a system user for the employee is created as well. Otherwise you can add employees as system users through the user management option under Admin tab. You may think that the fields for employee details are too few to capture all the details, but as you click Save after filling the details, you can see the full view of the employee details page. This page lets you view and modify all the details related to an employee like personal details, contact info, dependents, salary, organisational hierarchy and much more. You can also add custom fields under employee details page – just go to the Configuration tab under the PIM section and click Custom Fields. There are a few option fields available as well.



06 Time writing – timesheets



04 Leave management

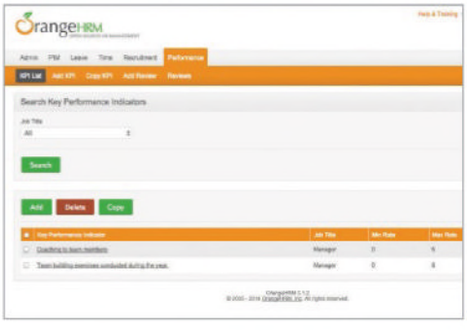
The next section is the Leave section. When you go to the Leave tab, new subsections are visible: Entitlements, Reports, Configuration, Leave List and Leave Assignment. To set up Leaves, you can start with the Configuration tab. Here you can create leave types (like sick, casual and more), list out holidays for the year, configure working days of the week and set the leave period (one year durations with leave entitlements are valid). After Configuration, you can then head over to the Entitlements section; this is the place to add leave entitlement to employees. Based on these entitlements and corresponding leave balances, employees can then apply their leaves. The Leave List section shows the leave data for the total leave period. The Assign Leave section allows the admin to grant leave without the employee applying for it.

05 Time writing – attendance

Attendance tracking is a very sensitive issue; even a small error in logging the in and out times can cause big problems (at least for the reputation of an employee). OrangeHRM provides a neat way to track the attendance. Just head over to the Time tab and you can see the Attendance section inside. Here you can view employee attendance records or configure things like whether or not an employee or supervisor can modify the attendance records. Note that you're currently logged in as admin, but to log attendance then you should be logged in to the system as an ESS user. Just make a user with the ESS role and log in. After that, just go to Time>Attendance>Punch in/out and click In. The system automatically logs the date and time. The page refreshes to show the Out option now.

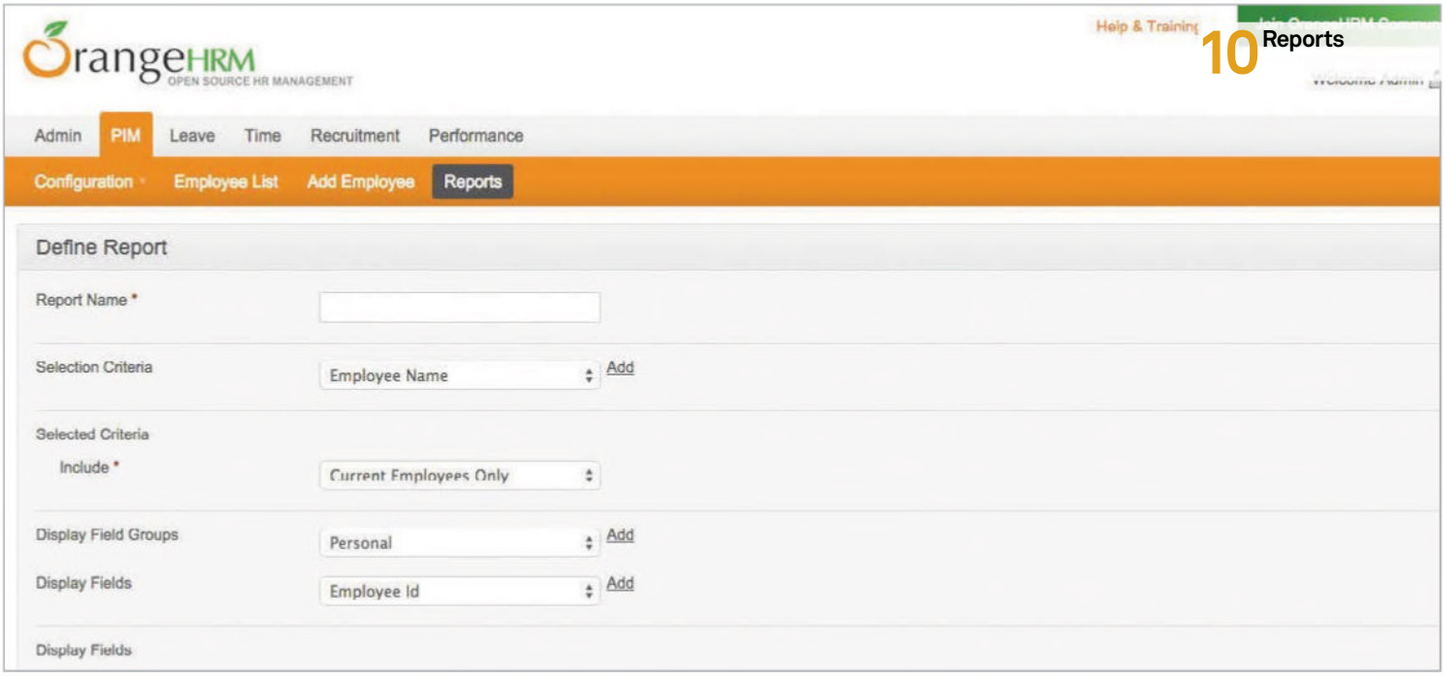
06 Time writing – timesheets

While attendance is used to track the actual hours spent in office, timesheets generally track the time spent on various activities inside the office, and the data is used for budgeting purposes. So before adding timesheets, you need to add customers and corresponding projects. Also, with each project the related activities should be added. To add these details, head over to Time>Project Info>Customers. After adding the customer, add projects using the Project link just below Customers – activities can also be added in the same page. Now you are ready to add timesheets. As an admin, you can add or view the timesheets of all the employees just by going to Time>Timesheets>Employee Timesheets. This page also shows the submitted timesheets, which have actions pending from your end. An ESS user can add and edit her timesheets using their own login.



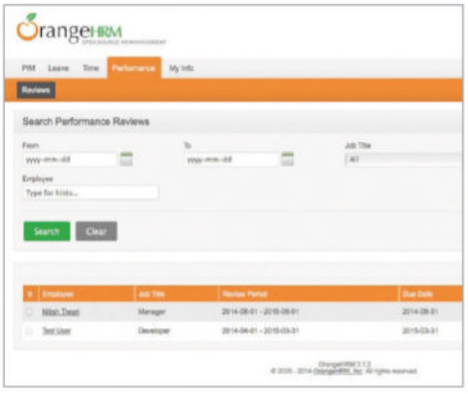
07 Performance management – employer

In this section we will have a look at how to set up performance management using the admin interface. The performance management in OrangeHRM is based on KPIs, ie the key performance indicators. For every job title in the organisation, the corresponding KPIs, along with the maximum and minimum rating points, need to be created. Every employee can then be automatically evaluated based on the KPI of their job title. To set KPI go to Performance>Add KPI. You can select the job title and then add the KPI; the maximum and minimum ratings are optional but should be added to ensure uniformity in ratings. You can check all the KPIs added in the KPI List page. After



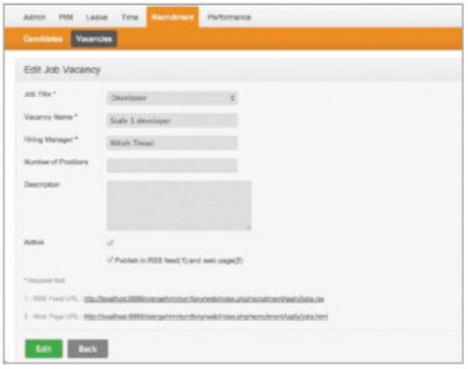
10 Reports

adding the KPI you need to go to the Add Review page to initiate the review process for an employee. In the next section we see how it all looks to the employees.



08 Performance management – employee

After the review process is initiated by the admin, the reviewer and the reviewee can view it under Performance>Reviews link. Only the reviewer can open it though, to add their reviews. After the reviewer adds reviews and ratings, and submits the form, the admin then needs to approve the review before it is made available to the employee. Once the admin approves the review, no further changes can be done – even by the admin. Note that the reviewer for an employee is completely independent of the organisational hierarchy, so anyone from the employee pool can be added as a reviewer for the employee.



09 Recruitment

OrangeHRM lets you manage the recruitment process as well. From publishing a vacancy to handling job applications to shortlisting and hiring – you can do it all with OrangeHRM. Let's see how to get started. Click on the link Recruitment>Vacancies, this is the page where you can add the vacancies. As a vacancy is created, a web link and a RSS feed is created, which is available publicly. This link not only has the full vacancy details but also allows candidates to apply via a form. Later, as someone applies, the candidate page (next to vacancy link) automatically gets updated with the details. You can then click on the candidate name to manage the application. The application goes through the steps of shortlisting, scheduling interviews, interview results, offering the job and, finally, hiring. After you hire the candidate, the employee entry for the candidate is created automatically. You would need to manually create the login for the new employee though.

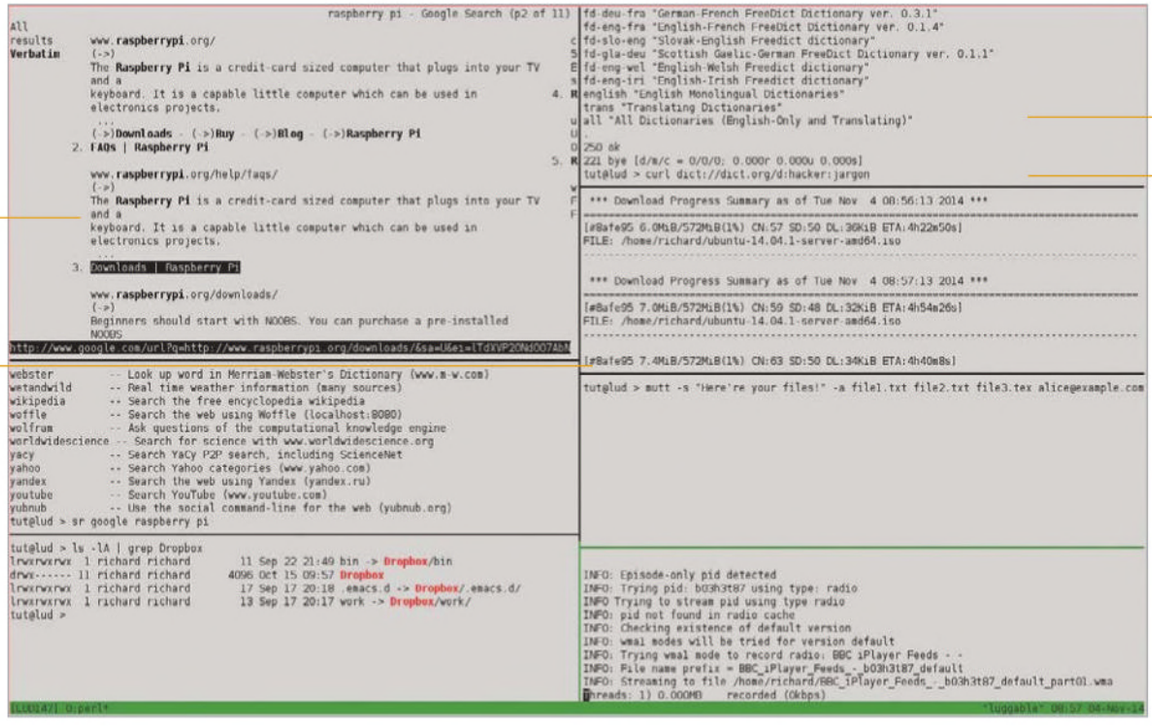
10 Reports

Several modules of OrangeHRM have the option to generate reports. Let's have a closer look at what each one of them has to offer. In the PIM section, the Reports tab lets you create custom reports which once created can be saved and used later. As you click on Reports the Define Reports page appears. Here you can create reports related to employee data such as employee's name, grade, job title, education and so on. The Leave section allows you to generate reports related to leave entitlements and leave usage. There are no custom reports here though, only a fixed set of reports. The Time section also has its own set of fixed reports. Project Reports shows the time spent on a project and its activities by all the employees, whereas Employee Reports shows the time spent by an individual employee, categorised by different projects and activities. Then while the first two are related to timesheets, Attendance Summary lets you see the attendance details of an employee.

"The performance management in OrangeHRM is based on KPIs"

Surfraw used the command line to search over a hundred engines and resources, leaving the browser for browsing

Great for scripting downloads, cURL will upload, grab headers or do dictionary look-ups from the command line



Most people migrated from command-line mail decades ago, but it's still there if you need to quickly attach a file

aria2 handles torrents, as well as downloading, from a variety of different sources

Use the Web from the terminal

Advisor

Richard Smedley
A Unix jack-of-all-trades, Richard always has a shell open so learnt scripting by osmosis. It's not that he dislikes GUI apps. He just loves the command line. A lot.

Resources

Surfraw surfraw.alioth.debian.org
cURL curl.haxx.se
wget www.gnu.org/software/wget
aria2 aria2.sf.net
youtubedl http://rg3.github.io/youtube-dl
get_iplayer www.squarepenguin.co.uk
get_flash_videos bit.ly/1yqzmzz

Browsers are great, but the command line saves time when searching, downloading and communicating on the Internet

From almost every app being on the command line to doing everything through the web browser, GNU/Linux has come a long way towards user-friendliness. But in always using that ever-present Firefox or Chromium session, something has been lost along the way.

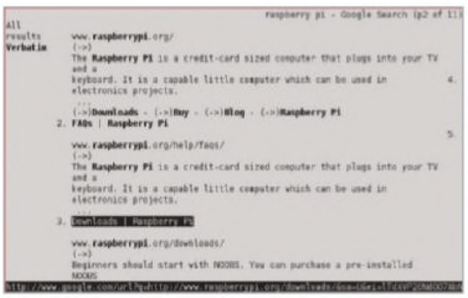
Every tab opened on the browser is time wasted in mouse operations and in seconds ticking away for the World Wide Wait for a AJAX-heavy page to load. Just as many GUI apps have arguably better equivalents on the command line, so too do many daily operations you carry out on the web have quicker terminal equivalents that can save time.

We're not just talking about saving a couple of seconds; going from a SSH session, checking logs on your server, to opening a web browser for a search on something involves moving concentration away from your project, as the sight of all of your open tabs beckons you to a multitude of distractions.

Remember, this isn't about replacing GUI apps with terminal ones – we're not covering browsers and IRC clients here; it's about getting things done on the web with a quick command in your terminal. We'll cover downloading and sharing, but let's start with where commands should be a natural fit: searching the web.

01 Before WikiLeaks

Surfraw stands for the Shell User's Revolutionary Front Rage Against the Web, and was written by Julian Assange many years before he became better known for another project. Surfraw is installable through your package manager and it will bring web searches to the command line.

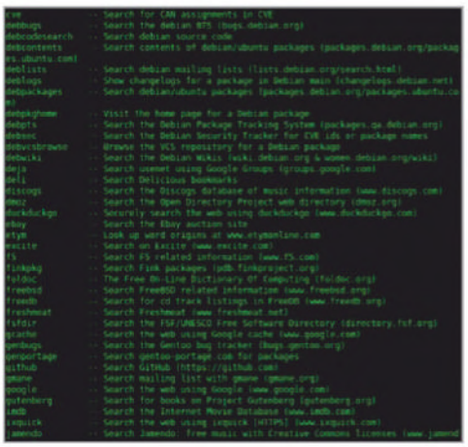


02 Surfraw

Putting search on the command line is a good fit, as you simply put:

```
sr google raspberry pi
```

...and you'll be looking at Google search results for Raspberry Pi in a sensible default browser (w3m on most Ubuntu systems). Other command line, or GUI, browsers can be set in the config file (note: all file locations given may vary depending on the distro).



03 Elvi search scripts

You can see more than a hundred available search options with:

```
sr -elvi
```

Elvi are the search scripts for various engines or sites. You'll find them in /usr/lib/surfraw/ and they, as well as surfraw options and arguments, are tab-completable.

04 Changing web

While some defaults are growing out of date – the late, lamented ntk and freshmeat feature are just two examples – Surfraw is still ready to go with many still useful search directories and is still being updated, with GitHub and jQuery docs among those added in the last release. Creating your own is left as an exercise for the reader.

05 Def and defyn

The commented config file is /etc/xdg/surfraw/conf – def and defyn are used here to define variables. The latter defines Boolean values such as:

```
defyn SURFRAW_graphical no
```

You can create per-user scripts in ~/.config/surfraw/conf with sh-style entries:

```
SURFRAW_graphical=no
```

06 In your script

The other side of the command line is shell scripting, to chain together utilities in repeatable programs. For this, Surfraw has a -p option to pass the URL to STDOUT instead of the default browser and an -o option to specify a text file to dump the browser's html.

```
sr -p rhyme -method=perfect orange
```

07 Get Wget

You've probably used GNU Wget before to grab a particular file or binary resource from a remote server. Add the -O option to specify a destination:

```
wget -O ~/bin/dropbox.py "https://www.
dropbox.com/download?dl=packages/dropbox.
py"
```

08 Fetch and clone

The two most useful options are -c, to resume an interrupted download (even one started by another program), and -r, which is a recursive fetch to a default depth of five directory levels, enabling you to fetch or clone whole websites.

09 Tips and tricks

Wget may be more primitive than the two rivals on the next page, but you'll find many Wget tricks for working around blockages to downloads, so you can grab a particular resource from, say, your command-line-only server. The -e switch enables many useful commands:

```
wget -e robots=off
```

10 cURL fetching

Handy as Wget is, cURL is a far more flexible fetching friend and it sends too. It's very invaluable for quickly checking the state of your sites with:

```
curl -I  gonetoearth.org
```

curl -I passes the headers of a site to the terminal.

11 Two-way street

cURL writes by default to STDOUT, which is handier for scripting, but -O will save the resource and a lowercase -o lets you specify a name to save as. When you're directing the output away from the terminal, cURL displays a progress meter there.

Credentials can be passed with -u to both http and ftp sites, and uploads to the latter made with the -T switch.

```
curl -u username:password -T "{file1,file2}"
ftp://ftp.myserver.com -T {"patch1,module1"}
ftp://ftp.mywebserver.com
```

Curl -X lets you specify PUT or POST methods instead of GET, for testing site features, even multipart forms.

12 Change the MOTD

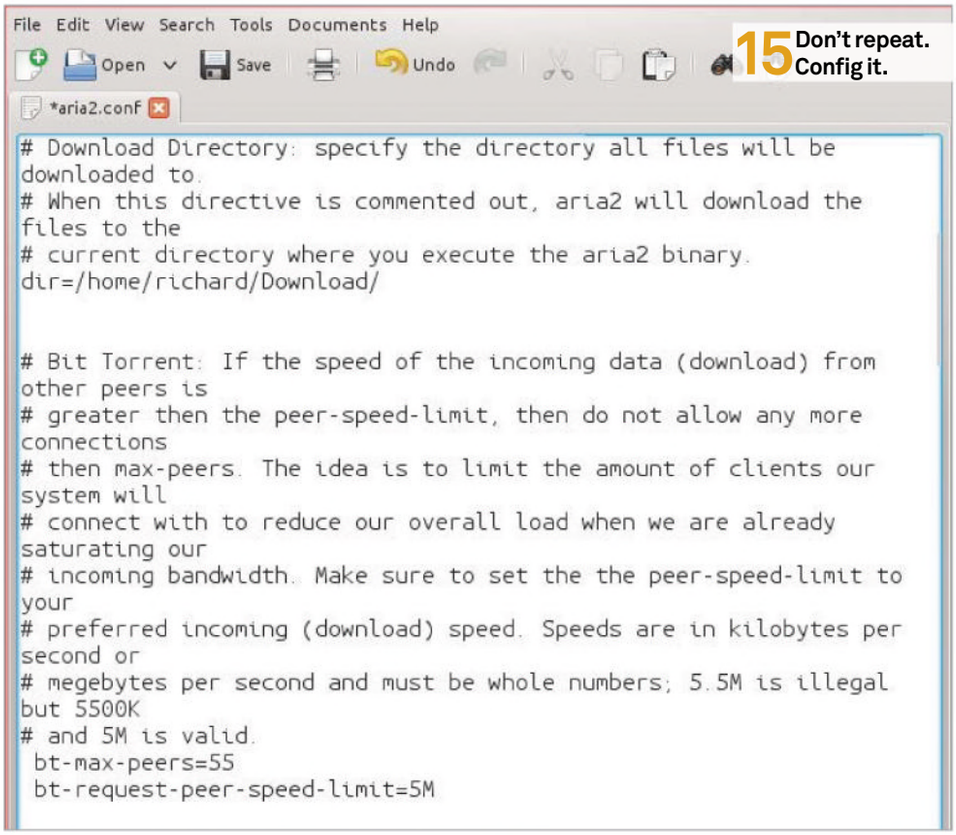
Looking for a change from your distro's usual MOTD (the message that greets you upon login)? Let cURL grab you a headline, joke or anything else from the the multitudinous resources of the web.

This command, for example, courtesy of **bashoneliners.com**, will give you a randomised string of corporate management jargon which may well be indistinguishable from recent communiqués from your bosses:

```
curl -s http://cbsg.sourceforge.net/cgi-
bin/live | grep -Eo '^<li>.*</li>' | sed
s,\</\\?li\>,,g | shuf -n 1
```

```
aria2c --seed-time=120 --seed-ratio=1.0
```

"Aria2 works with torrents, which remain the best way for downloading distros"



13 Grab with aria2

Wget is installed by default almost everywhere and cURL is attaining default status too. By contrast, aria2 is not so well-known, but it's a good way of grabbing the latest ISO – or any file or software, as metalink tries to look for the best version by location, language and OS.

14 Share the (down)load

Aria2 works with torrents, which remain the best way for downloading distros. Everything from upload throttling to share ratio can be specified on the command line.

```
aria2c --seed-time=120 --seed-ratio=3.0
http://releases.ubuntu.com/14.04.1/ubuntu-
14.04.1-server-amd64.iso.torrent
```

15 Don't repeat. Config it

Aria2's config file saves you retyping command line options such as where you want downloads placed, the rate limits for torrents, and the log level.

Uncomment and change the defaults as needed – but if your distro doesn't install a config file set your own:

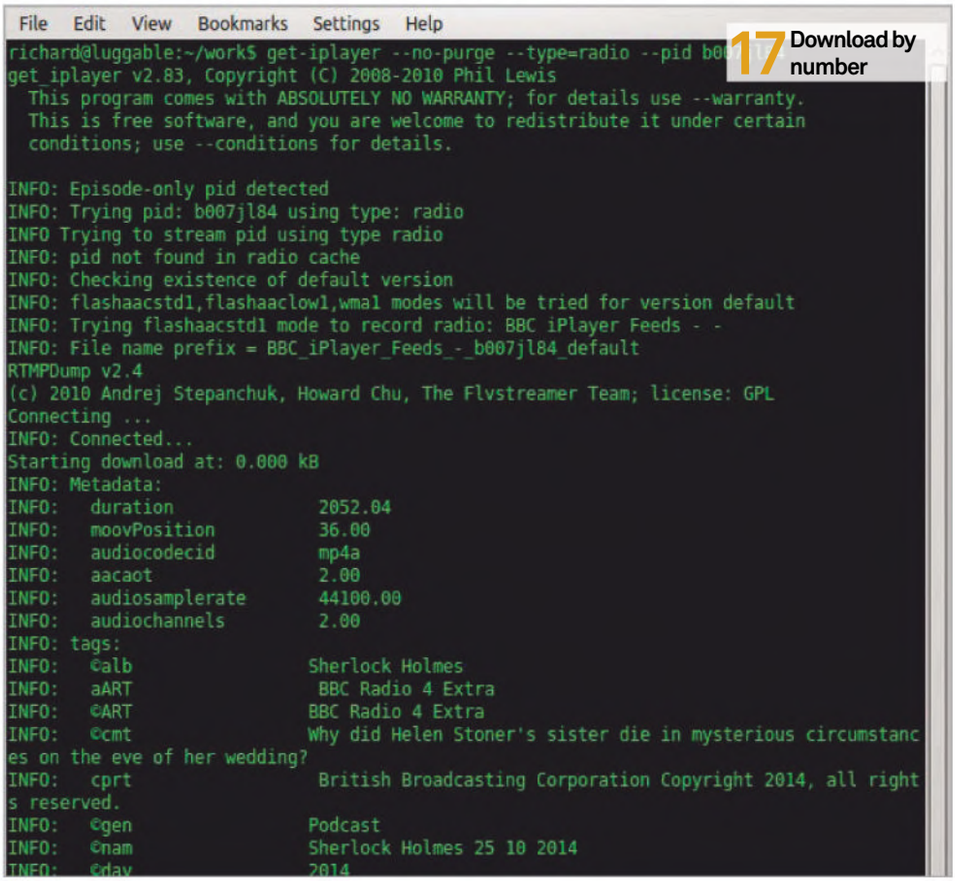
```
log-level=warn
max-connection-per-server=4
min-split-size=5M
on-download-complete=exit
listen-port=60000
dht-listen-port=60000
seed-ratio=2.0
max-upload-limit=50K
```

Whether aria2, Wget or anything else, using the same options twice is a strong hint that you should start to open up the config file and set some sensible defaults for your most common actions.

16 On the Beeb

Get_iplayer is a handy little Perl script that, almost since the launch of the BBC's iPlayer service, has brought programme catch-up to non-x86 platforms and those without a fast enough connection to stream in real time.

It occasionally has to play catch-up with changes to the service and, as we go to press, the BBC has dropped the programme data feeds that gave get_iplayer search and PVR capabilities. See **squarepenguin.co.uk** for any updates on this.



17 Download by number

Get_iplayer still works with the pid you see embedded in the iPlayer web page URI for each programme you might want to download, so although you'll need to browse the website until there's a workaround, you can at least grab a programme like this:

```
get-iplayer --no-purge --pid p01x5k4n
```

18 YouTube downloader

YouTube is a massive knowledge repository, containing instructional videos on everything from Beagle Boards to natural swimming pools (ie big ponds). They're great for a long train journey where an intermittent Internet connection would make life difficult. Download ahead of time with youtube-dl (which also works with some other sites); just feed it the URL:

```
youtube-dl http://youtube.com/
watch?v=za8FMIWYtUc
```

If older versions give a 403 error, update or change https to http in the command, as above.

19 Flash without the web

Get_flash_videos will usually help on sites where youtube-dl fails, but not always. With both apps, get into the habit of double-quoting URLs, so the shell doesn't try and interpret special characters like &.

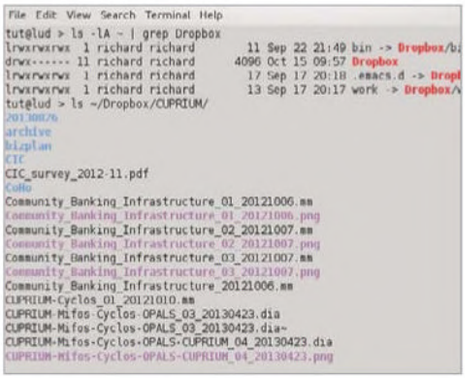
20 Shared storage and cloud services

Free and open cloud services are appearing with the burgeoning IndieTech movement, but Dropbox is still the service that most of us have accounts on – particularly as we often have to share files with other users for work. It's a reasonable place to keep extra copies of config files you share across machines, for example.

The command-line Dropbox script, which starts the service with Dropbox start, saves you running the resource-hogging Nautilus. Use symbolic links to save from disrupting your normal file hierarchy:

```
cd ~
mkdir Dropbox/.emacs.d
ln -s Dropbox/.emacs.d
```

Avoiding Dropbox and others with proprietary



components usually means setting up your own Cloud server, but Seafile, which is aimed at collaborating teams, offers 1GB free at **seacloud.cc**. It also offers software for your own server. Seafile is hosted on Amazon Web Services and written in Python; it's well worth comparing with other 'own cloud' solutions.

21 Mail servers

We're browsing, downloading and sharing without the Browser, but don't forget command-line email goes back decades before the web. Mutt is still one of the most efficient mailers out there - whether you're on Gmail, or your own mail server.

Whether you're using the built-in mail (you may need to install mailutils) or go with Mutt, the syntax is similar:

```
mail -s "Hello, World!" hi@gmail.com <body.
txt
```

22 Browser commands

If you like the power of the commandline, but really spend more time in a browser than a terminal, try **YubNub.org** - a command-line-style web interface to search engines and more. Check out **yubnub.org/kernel/most_used_commands** to see the most popular of the tens of thousands of user-contributed commands.

Real-time log monitoring with Swatch

Get notifications from predefined log events by setting Swatch to monitor for certain keywords

Advisor

Swayam Prakasha
has a master's degree in computer engineering. He has been working in IT for years, focusing on areas like operating systems, network security and electronic commerce

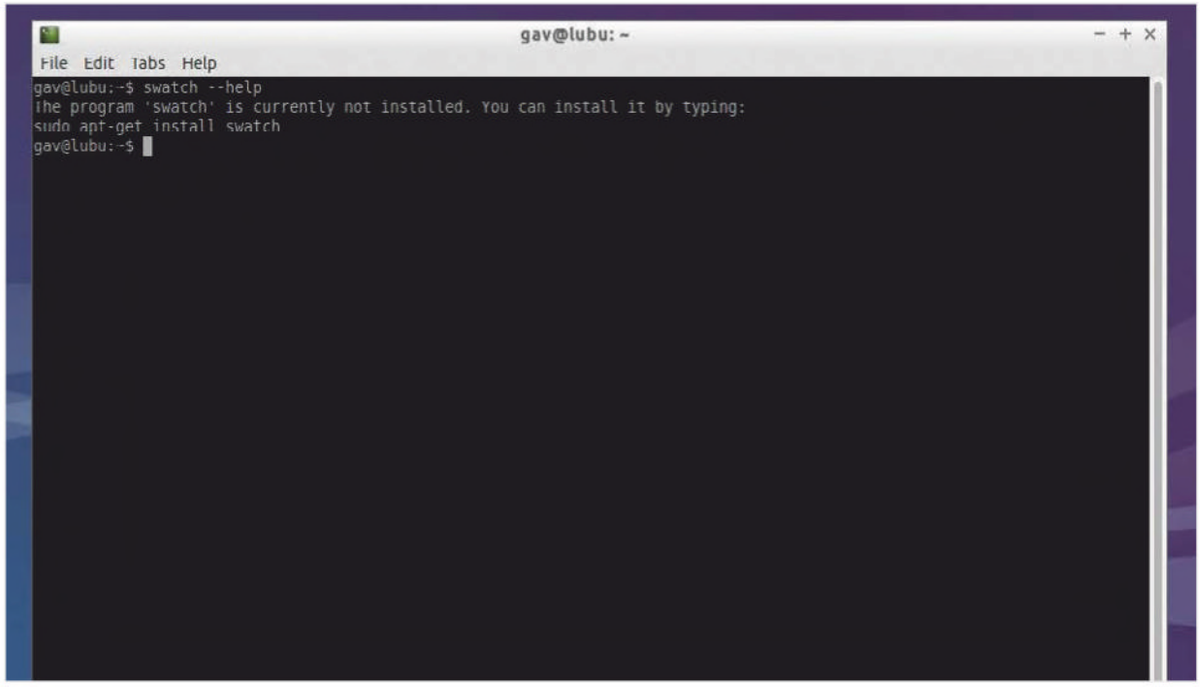
Resources

Swatch bit.ly/1KXNNDb

Swatch stands for Simple Log Watcher or syslog watcher, depending on whom you ask. Either way, Swatch is a helpful program that does your log watching and notifies you only when things that you are specifically looking for get logged. Note that Swatch is a Perl program that regularly sweeps the main log files and looks for certain keywords that you can define. It can be run in two ways: in the background as a daemon or as a cron job. You can configure Swatch to alert you of any events in the messages or syslog log files that might indicate a security problem. However, Swatch can also be used to flag just about any kind of activity: a certain program being used, a particular user logging in or anything else that might appear in a log file. Swatch can be configured to watch application-specific log files instead of the general log files that it does by default. Swatch is a Linux tool that helps in monitoring the log files as they are being written to. It then takes necessary action if it finds something that it is configured to look for. This tool can be used as a way to proactively scan log files in real time for various suspicious activities, error messages or specific keywords.

In brief, Swatch basically started out as a simple watchdog for actively monitoring the log files produced by UNIX's syslog facility. Since then, it has been evolved as a utility that can monitor just about any type of log. You can consider Swatch as a command line utility that can be started by issuing a **swatch** command with various settings following.

Please note that certain events that are logged have a great significance from a security standpoint. The default items that Swatch looks for are a good start:



You may already have Swatch installed, so check with **swatch --help**

- **Bad logins**: when the words 'invalid', 'repeated' or 'incomplete' appear in the messages file
- **System crashes**: when the words 'panic' or 'halt' appear in the log files
- **System reboots**: the banner of your OS should only appear in the log files when you reboot

Note that Swatch requires Perl 5 or higher. If you have a fairly new installation of Linux or BSD, then you should have a sufficiently current version.

Swatch requires multiple Perl modules to be installed in order to function correctly. You must first install CPAN and download each module via the CPAN console. In order to install these modules, you may be prompted to install additional modules as well – the configuration process will tell you if you are missing any of these.

We will need to use the following command to ensure that the required Perl modules are installed:

```
cpan -i module-name
```

...where **module-name** needs to be replaced with **Date::Calc, Date::HiRes** and then **Date::Format**. Download the tar file from the SourceForge website and unzip it.

Use this command to extract the files:

```
$tar -zxvf  swatch-3.2.3.tar.gz
```

Since Swatch is a Perl program, the installation process is slightly different to usual. Here is the sequence of commands that you'll need to follow:

```
perl Makefile.PL
make
make test
make install
make realclean
```

Once these processes are done, Swatch is installed and you are now ready to go.

After installing Swatch, you will be interested in creating a configuration file. If we look at the contents of the Swatch configuration file, you can see that the syntax is very simple. All it requires is a definition of what to search for followed by an action for if a specific match is located. It is important to note that Swatch utilises Perl regular expressions to define the search parameters and perform a variety of actions (turn to page 56 for more on regular expressions).

First, you need to create an empty file to be used as a configuration file. The normal practice is to create this file under /etc, and you can modify it by using any basic text editor. The Swatch configuration file is where you'll find all of the important settings, and inside this file, called swatchrc by default, you can tell the program what to look for in the log files and what to do if it shows up. Since the whole point of Swatch is to simplify our lives, configuring Swatch is pretty simple because everything is controlled by that single file: $HOME/.swatchrc (by default). It contains text patterns in the form of regular expressions that you want Swatch to watch for. Each regular expression is followed by the action(s) you wish Swatch to take whenever it encounters that text.

The configuration file syntax begins with a **watchfor** keyword. Basically, this line instructs Swatch to search for specific patterns in the form of regular expressions. Each **watchfor** line is then followed by an action.

There are two options available in swatchrc to specify the patterns to look for. They are:

- **Watch for regex** – an appropriate action will be taken when the regular expression specified in regex is found within the file or command being monitored
- **Ignore regex** – take an action when there is any activity within the file or command being monitored, except for events that match the expression specified in regex

In the following example, we will simply output the log entry to the console if an SSH connection has been established by issuing the **echo** action.

```
watchfor /ssh/
      echo bold
```

Let us take a quick look at some of the options that are available with Swatch:

Option	Description
--config-file filename	Run Swatch using the specified filename as configuration file. If no filename is given, use the default one
--restart-time time	Restart Swatch at the indicated time
--daemon	Run Swatch as a system daemon
--examine file	Make Swatch do a complete pass through the specified file
--tail-file	Swatch reads only the newly added ines in the file
--help	Display a short help summary
--version	Display the version of Swatch script

For example, the following command:

```
./swatch --config-file /home/swayam/my-swatch-
  config -- daemon
```

Sample setups

Generally, developers come across typical scenarios where they need Swatch to monitor situations like unexpected restarts. A typical configuration would be watchfor /halt/restart/panic. You then add actions such as ringing the PC speaker using the bell keyword and sending an email alert to the system administrator.
Also, be sure to specify multiple email addresses for different people whenever there is a critical watchfor section, like this:

```
watchfor /ssh.*failed/
{
mail=admin1@company.
com;admin2@        company.com
}
```

...will run Swatch using the configuration file found at /home/swayam/my-swatch-config instead of the default configuration file. It will also run it as a background process or daemon. Please note that the above options may be issued alone or together.

Swatch expects the .swatchrc file to live in the home directory of the user who invokes the program. Swatch also keeps its temporary files there by default. Each time it's invoked, it creates and runs a script called a watcher process, whose name ends with a dot followed by the PID of the Swatch process that created it. However, Swatch generally doesn't clean up after itself very well, instead it tends to leave watcher-process scripts behind. Consequently, users are expected to keep an eye out and periodically delete these in their home directory.

The command **--tail-file=[path to log file]** directs Swatch to watch a specific log file for potential matches.

An interesting feature of Swatch is that you can run multiple instances of Swatch, each configured to use a customised configuration file and watch a different log file. Some of the common log files that Swatch can use have been listed here below.

- **/var/log/maillog** – logs all email messages
- **/var/log/cron** – logs messages about cron job schedules

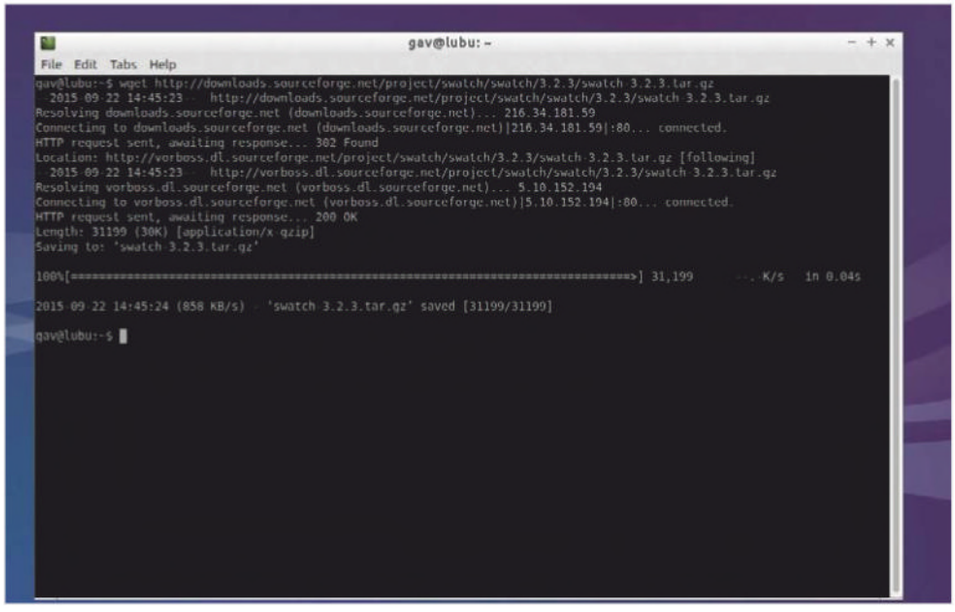
- **/var/log/messages** – logs system messages

Let us understand all of the actions that Swatch can take when a specific search is found. Swatch provides a variety of actions that you can perform in response to a matched event. You can output alerts to the console, pipe output to another log file, send email alerts or even execute a remediation script. More than one action can be applied to a single event, therefore you can combine these to suit your personal requirements. The following table illustrates various Swatch action statements:

Action Statement	Description
echo [mode]	The search text can be echoed onto the screen. Note that **mode** is optional and indicates the colour in which it is to be displayed, eg **echo magenta**
bell [number]	This rings the PC internal speaker the number of times indicated
exec [command]	Executes a command line parameter. You can configure this to call a script that can then take further action
pipe [command]	This passes along a command to another process
write [user1]:[user2]	It causes an alert to be sent via the UNIX **write** command and can be sent to one user or a group of users
mail	Sends an email using the Sendmail
addresses=[address1]:	program to a single email address or
[address2]:	multiple email addresses that are
[add3],subject=[text]	separated by colons
throttle HH:MM:SS	Wait for HH:MM:SS (period of time) after a line triggers a match, before performing actions on another match of the same expression

As can be seen from the table, Swatch can notify you of flagged log events in several different ways. The easiest is to have it beep or echo on the screen. If you are not around the server all the time then you can have it email you. If your pager or cell phone supports text messaging via email then you could have it send the message directly to you. You can also write a script to have the server dial a pager number using the UNIX tip command.

Echoing the log output is considered one of the most basic functions of Swatch. This can be utilised as a way to gain the attention of the user (if they are currently using the console) by outputting the log contents to the console.



Swatch is hosted on SourceForge, so check there for the latest package

To utilise the echo action, simply issue the **echo** keyword underneath the **watchfor** line as follows:

```
watchfor /su|sudo/
    echo [formatting keywords]
```

Although it is very basic, we can note here that echo offers a variety of formatting methods in which to display alerts. You can set the text colour, underline, bold, strike-through, flash text, as well as combine multiple formatting keywords.

Swatch can alert a user when there is a positive match by issuing a bell sound. The following example illustrates this:

```
watchfor /su|sudo/
    bell [total # of rings]
```

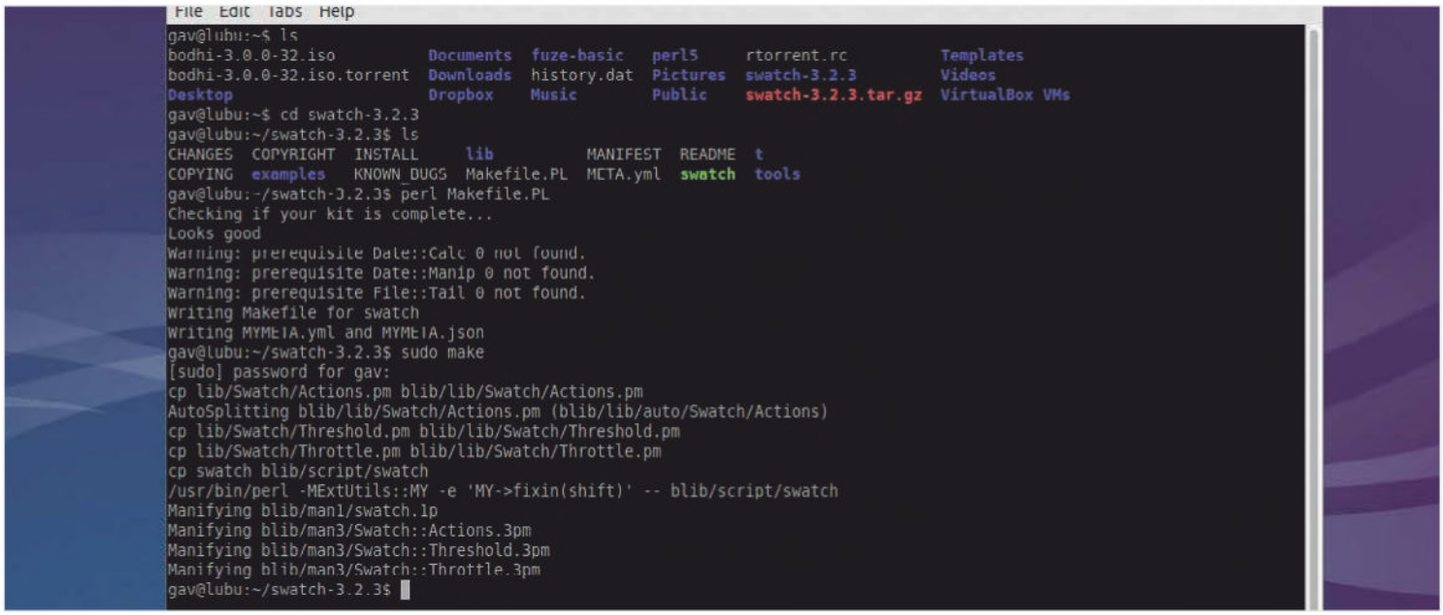
In what is considered a more efficient way of alerting, you can configure Swatch to send you an email whenever there is a match for a specific event. This can be considered a convenient way to alert a system administrator of real-time events without requiring them to be at the console. Let us take a look at an example: we will be sending an email to the administrator if a new application has been installed on the system. Note that the address contains an escape character for the email address \@ – this is required for Perl to format the address correctly. Also, if you would like to send spaces within your subject line, you must place an escape character prior to each space:

```
watchfor /[iI]nstalled/
    mail addresses=admin_person\@mycompany.
com,
    subject=Unauthorised\ Application\
Installation
```

It is important to note here that the **throttle** action helps to prevent denial of service attacks via Swatch (eg deliberately triggering huge numbers of Swatch

Swatch in summary

Although Swatch is limited in its abilities, it proves to be a very powerful tool to implement alongside other security products to proactively monitor system logs. The goal of a system administrator should be to run Swatch neither 'too hot' (alerting us to routine or trivial events) nor 'too cold' (never alerting us about anything). Swatch gives system administrators great log monitoring options; as a result, it's a perfect tool for monitoring SSH or Denial-of-Service attacks on your Linux servers, possibly alerting you to trouble before it's too late.



You'll see lots of prerequisite warnings as you run the **make** steps

events in a short period). In other words, throttle gives Swatch the intelligence to ignore repeated occurrences of a given event, potentially preventing minor events from becoming major annoyances.

As a part of remediation, Swatch has the ability to execute a secondary script if a specific event has been detected. This functionality of Swatch could trigger a further action or actions in response to an event. The syntax required you to use the **exec** keyword and accepts some bash, Perl and other very useful commands.

The following example illustrates this as we direct Swatch to execute a remediation Perl script if it detects a port scan against the system:

```
watchfor /[cC]onnection closed by/
    exec "perl /usr/bin/custom_remediation.pl"
```

Another important scenario where Swatch can be used is to avoid a storm of alert messages. That is, we can configure Swatch to take a specific action only if it detects a certain number of similar events within a certain timeframe. Let us consider a scenario where there are multiple SSH failed log-in attempts within a specific time period. We can configure Swatch to search for this case and take an action, sending a message to the system administrator as well as outputting a log message to the console. The following command will handle such a case:

```
watchfor /ssh.*failed/
    echo bold
    mail addresses=admin_person\@mycompany.com,
    subject=Possible\ SSH\ Brute\ Force\ Attempts
    threshold track_by=$1, type=threshold,
  count=5, seconds=10
```

"A feature that Swatch provides is its ability to specify a given time of day an action will be performed"

We can note in this example that we have set the threshold here as five failed attempts to log in to the system within ten seconds, although you can of course set your own.

Another powerful feature that Swatch provides is its ability to specify a given time of day that an action will be performed. This will be very beneficial to perform a certain set of actions over the weekend or after business hours. To apply a time constriction to an action, we need to append the keyword **when=** followed by the time duration after an action. The syntax used to represent the timeframe is indicated in numerical format: each day of the week is represented by a number between one and seven (7 = Saturday, 1 = Sunday) and hours are represented in a twenty-four hour timeframe between one and twenty four. The following example illustrates this concept:

```
watchfor /system full/
    mail addresses=admin_person\@company.com,
    subject=File\ System\ Full,
    when=7-1:1-24
```

A user will be alerted by Swatch only over the weekend if a storage drive becomes full.

It is always advisable to follow some best practices when you are using Swatch. If Swatch's actions don't fire very often, it could be because your system isn't getting probed or misused very much. Nevertheless, it could be just as likely that Swatch isn't casting its net wide enough. In such cases, you may need to continue to periodically scan through your logs manually just to see if you're missing anything and then continue to tweak the Swatch configuration file .swatchrc. As another good policy, you should never forget to periodically reconsider the auditing/logging configurations of the daemons that generate log messages in the first place. It is critical to realise that Swatch won't catch those events that aren't logged at all.

Turn an old PC into a NAS box

Advisor

Phil King Since starting out on CRASH magazine in 1988, veteran videogames and technology journalist Phil has tinkered with all sorts of hardware and reviewed hundreds of apps and games.

Resources

Spare PC with at least 512MB of RAM

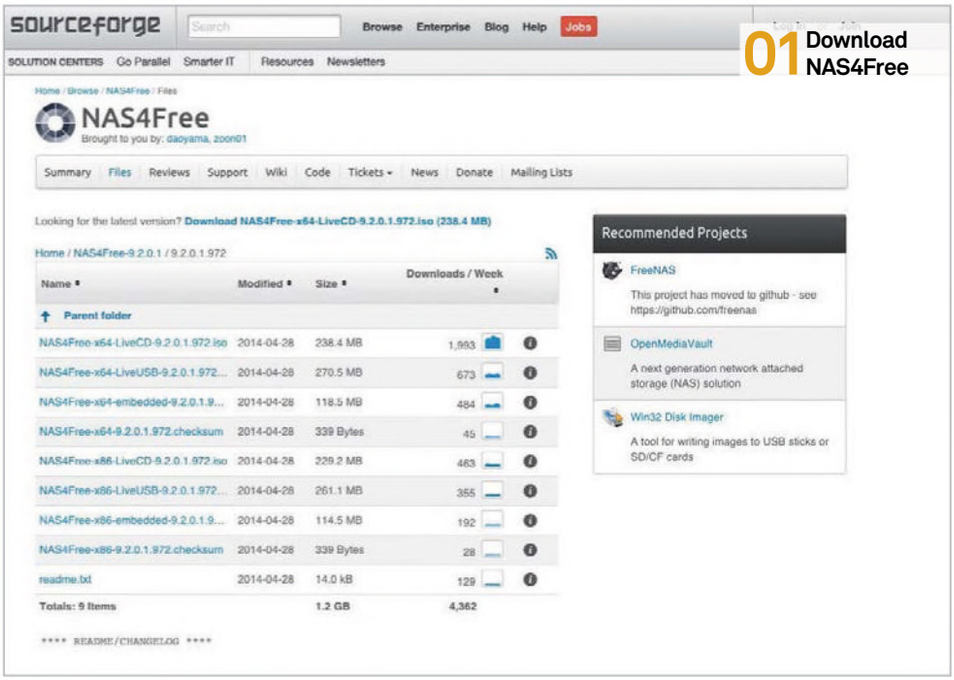
FreeNAS
sourceforge.net/projects/nas4free/files

Home network

Repurpose old hardware with NAS4Free to use as a NAS server for backups and more

The fast pace of technological progress is great, but it does mean that hardware soon becomes redundant. This begs the question: what do you do with that old PC gathering dust in the attic? Apart from selling it or giving it away, another option is to turn it into a network-attached server for storing files, media and backups. For this purpose there are several specialist distros to choose from, including FreeNAS and OpenMedia Vault. However, to encompass as much older hardware as possible, we'll be using NAS4Free – a legacy version of FreeNAS – since it has lower system requirements. Officially, it only requires 512MB of RAM to work, but you may be able to get away with as little of 256MB for the Full version.

We'll show you how to install NAS4Free on your old PC and then access and configure it remotely from a client PC via its web-based GUI. You can then schedule regular remote backups of selected folders using rsync and cron (or Windows Backup or OS X Time Machine). We also cover other uses including UPnP media streaming and downloading torrents (using the built-in Transmission) – you could even set up ownCloud hosting. So dust off that old PC and let's get it working for you again!



01 Download NAS4Free

01 Download NAS4Free

You can find the latest NAS4Free files at SourceForge. Choose either a Live CD ISO or Live USB IMG file, depending on whether you want to boot it from CD or USB. Also, select the correct version for your PC: x64 (64-bit) or x86 (32-bit).

02 Boot it up

After setting the BIOS on your old PC so it'll boot first from CD (or the USB stick), insert your live disc/stick and boot it up. NAS4Free will go through the boot process, which may take a while to complete.

03 Choose install method

You'll come to a Console Menu. Enter 9 to install from your live CD/USB. In the next menu, choose option 2 to install it on the PC's hard disk (or 1 if you want to run the OS from a USB flash drive).

04 Install to disk

Hit OK on the next menu, choose the installation media and destination media, then say No to a swap partition (unless you have very little RAM). NAS4Free will then be installed on the chosen disk. Note the DATA partition parameters.

05 Configure LAN interface

Now remove the live CD/USB and reboot the computer. After the bootup process, you'll end up back at the same Console Menu. This time, enter 1 to select your Ethernet interface (probably from just one option).

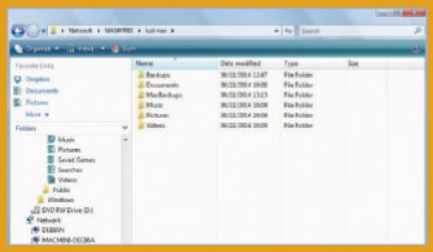
06 Configure IP address

Back at the Console Menu, enter 2 to configure the network IP address. Say No to DHCP and enter a static IP. Press Enter to accept the default subnet mask. Use your router's IP address as the default gateway and enter your favoured DNS.

07 Access web GUI

With the basic setup done, you can now access your NAS4Free server from another PC. Just enter its IP address in a web browser and you'll see the NAS4Free web GUI. The default username is 'admin', with password 'nas4free'.

Back up OS X and Windows



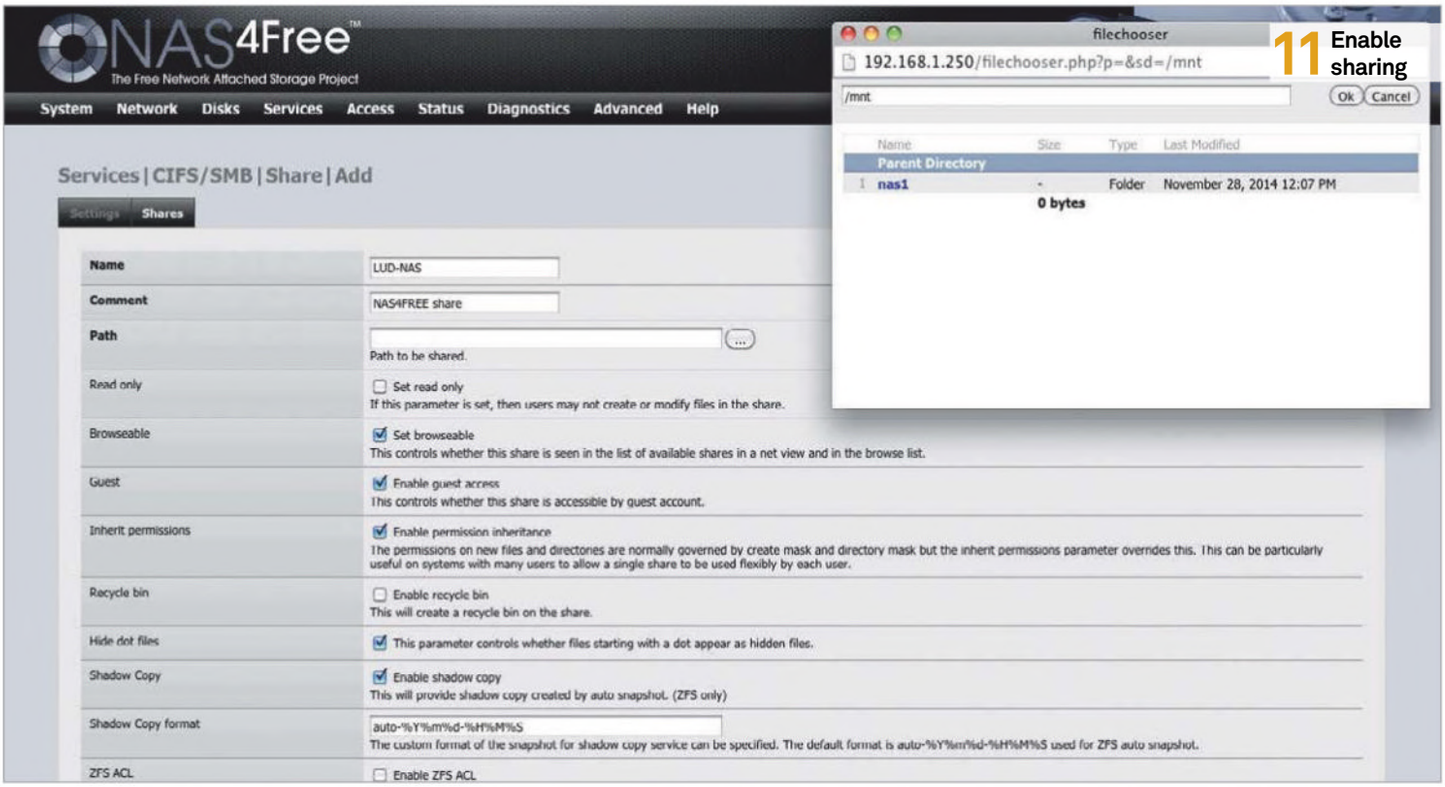
Easily back up a Windows PC

You can access your NAS4Free CIFS/SMB share on a Windows PC by typing \\[your NAS4Free IP address] in the Explorer. While you could back up using rsync, it's easier to use the Windows Backup feature (on Windows 7 Professional or later).

Go to Backup & Restore>Set Up Backup, then hit the 'Save on a Network' button. Browse to your NAS4Free shared folder, then click Next, choose backup settings and set the schedule for them.

Back up your Mac via AFP

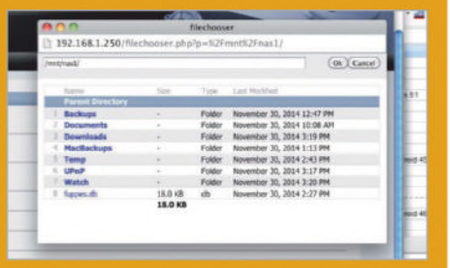
Again, you could use rsync, but to use Time Machine just share your NAS4Free drive via AFP. In the web GUI, go to Services>Users & Groups and click Groups. Click '+', fill the fields, then Add and Apply Changes. Click Users>+ and fill in the fields, assigning the Primary Group as your new one. Go to Services>AFP and click Shares. Click '+', add a name and comment, hit the Path '...' button and choose your drive's mount point. Enable automatic disk discovery and choose Time Machine. Click Add>Apply Changes. In Settings, click Enable, tick both authentication options, then Save and Restart. Now hit Go>Connect to Server and enter afp://[NAS4Free IP]. In Time Machine's Preferences, hit Select Disk and you'll see your shared NAS4Free folder.



11 Enable sharing

BitTorrent client

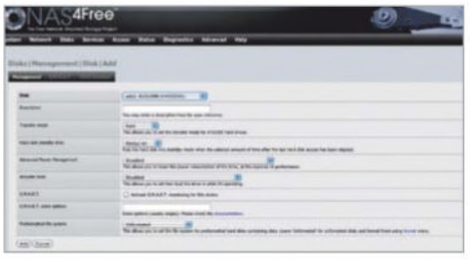
Let your NAS box handle all your torrents



Another neat feature of NAS4Free is its built-in Transmission BitTorrent client. From the web GUI, go to Services>BitTorrent and click Enable. Add the download and watch directories, alter any other settings you want, then hit Save and Restart. Now, whenever you add a torrent to the watch folder (from any connected PC), your NAS4Free server will start downloading it. Click the URL at the bottom of the Services>BitTorrent screen to check its progress. Note: you may need to get your router to forward the port used.

08 General settings

For extra security, you can change the username and password via System>General – click the Password tab to change it. The General menu also enables you to alter settings such as DNS and time zone.



09 Add disk

Go to Disks>Management and click the '+' on the right. Choose your hard disk from the drop-down, then the file system for a pre-formatted disk – if it's not, you can format it via Disks>Format. Click Add at the bottom, then Apply Changes.

10 Add mount point

You need to add a mount point for the disk. Go to Disks>Mount Point and click '+'. Choose your disk from the drop-down, keep UFS file system enter partition number 1 and then a mount point name. Click Add, then Apply Changes.

11 Enable sharing

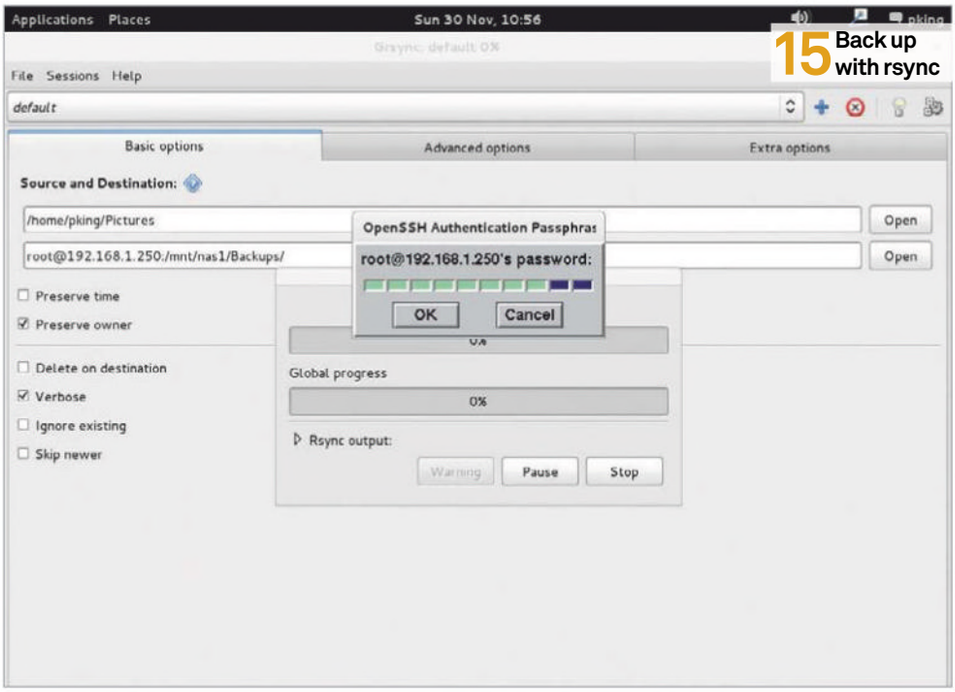
Go to Services>CIFS/SMB and click Enable. Click the Shares tab, then '+' and enter a name and comment. Click '...' for Path and choose your mount point name from the pop-up. Click Add, then Apply Changes. Click the Settings tab, then Save and Restart.

12 Remote access

You can now access the shared folder from the file browser of another PC – Browse Network>Windows Network>WORKGROUP>NAS4FREE>shared folder. Create a Backups subfolder in it, to separate them from shared files and media.

13 Set up rsync

On the web GUI, go to Services>Rsync. Click the Modules tab, then enter a name and comment. Hit the Path '...' button, select your mount point and Backups subfolder. Click OK, Add, Apply Changes. Click Settings tab, Enable, then Save and Restart.



15 Back up with rsync

"For extra security, you can change the username and password"

14 Start up SSH

We'll want to use rsync with SSH to back up files securely from the client computer to our NAS4Free server. In the web GUI, go to Services>SSH and click Enable. Tick the 'Permit root login' option. Then click Save and Restart.

15 Back up with rsync

Now let's try a manual backup from the client PC. While you can run rsync from the command line, we're using Grsync – a GUI front-end – for ease of use, particularly when choosing options. Choose the folder to back up, then enter the destination: root@[NAS4Free IP]:/mnt/[mount point]/Backups. Click the gears icon and a dialog will then prompt you for a passphrase: enter your NAS4Free password (default is 'nas4free'). The backup will then proceed. This is fine for manual backups, but for automated ones we'll need to set up SSH password-less, key authentication.

16 SSH login

Setting up SSH key authentication (see bit.ly/1zGfaug) is done from the command line. First, open a terminal and enter:

```
ssh -l root [NAS4Free IP]
```

Type 'yes', then enter the password to log in to your NAS4Free server.

17 Generate SSH key

Now we can generate a SSH key pair, just by entering:

```
ssh-keygen
```

Press Enter to accept the default file location, then Enter to set an empty passphrase and Enter again to confirm it. Your SSH key pair will then be generated.

UPnP streaming

Stream music, videos and photos

You can also turn your NAS backup box into a UPnP media server. Make a folder for your UPnP server on the shared disk, via a client PC's file browser or SSH, and subfolders for Music, Photos, Videos. Go to Services>DLNA/UPnP in the Free4NAS web GUI, click Enable and choose your new folder as the database directory. For the media library, click '...' and browse a subfolders, click OK, then Add. Repeat for the others. Choose a profile for your UPnP device. Enable transcoding if needed and select the Temporary directory.

18 Rename public key

Rename your public key with:

```
mv ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
```

Then log out with:

```
exit
```

Copy the private key to your client PC with:

```
scp -p root@[NAS4Free IP]:~/.ssh/id_rsa ~/.ssh
```

Enter the password, then SSH in (step 16) and you won't be asked for a password.

19 Automate backups

You can now automate backups with cron. In the terminal, enter:

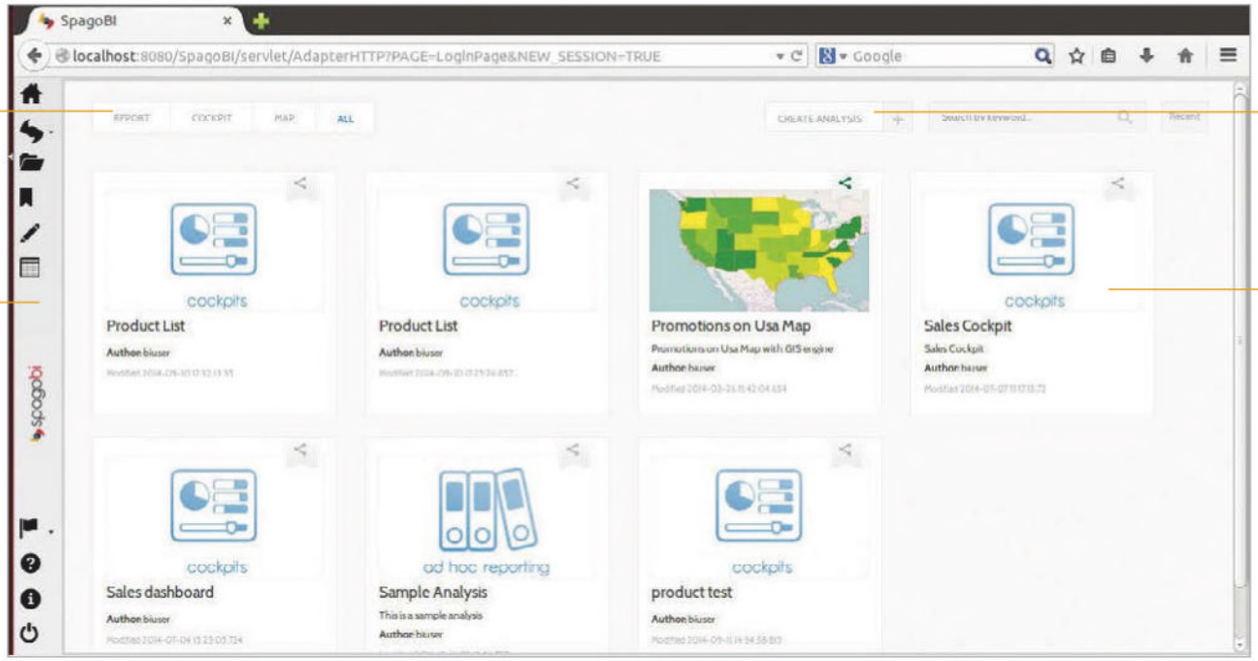
```
crontab -e
```

Copy and paste your rsync command into a new line at the bottom, preceded by the time and date fields – it's mins, hour, then * * * for a daily backup at that time, so for a 2pm daily backup you'd use:

```
00 14 * * *
```

These buttons serve as a filter for saved analysis. Since All is selected, it's showing the available types of saved analysis

This creates a new analysis. Click and you can select the analysis type, including ad hoc, cockpit or location based



This is the menu bar for regular users. Starting from home to the logout button below, it is available in all pages

All the analysis that's saved for viewing later is available here. You can simply click the blocks to view a particular analysis

Get key insights from business data with SpagoBI

Businesses need to ensure they make the right decisions based on their data. This is where business intelligence tools come in

Advisor

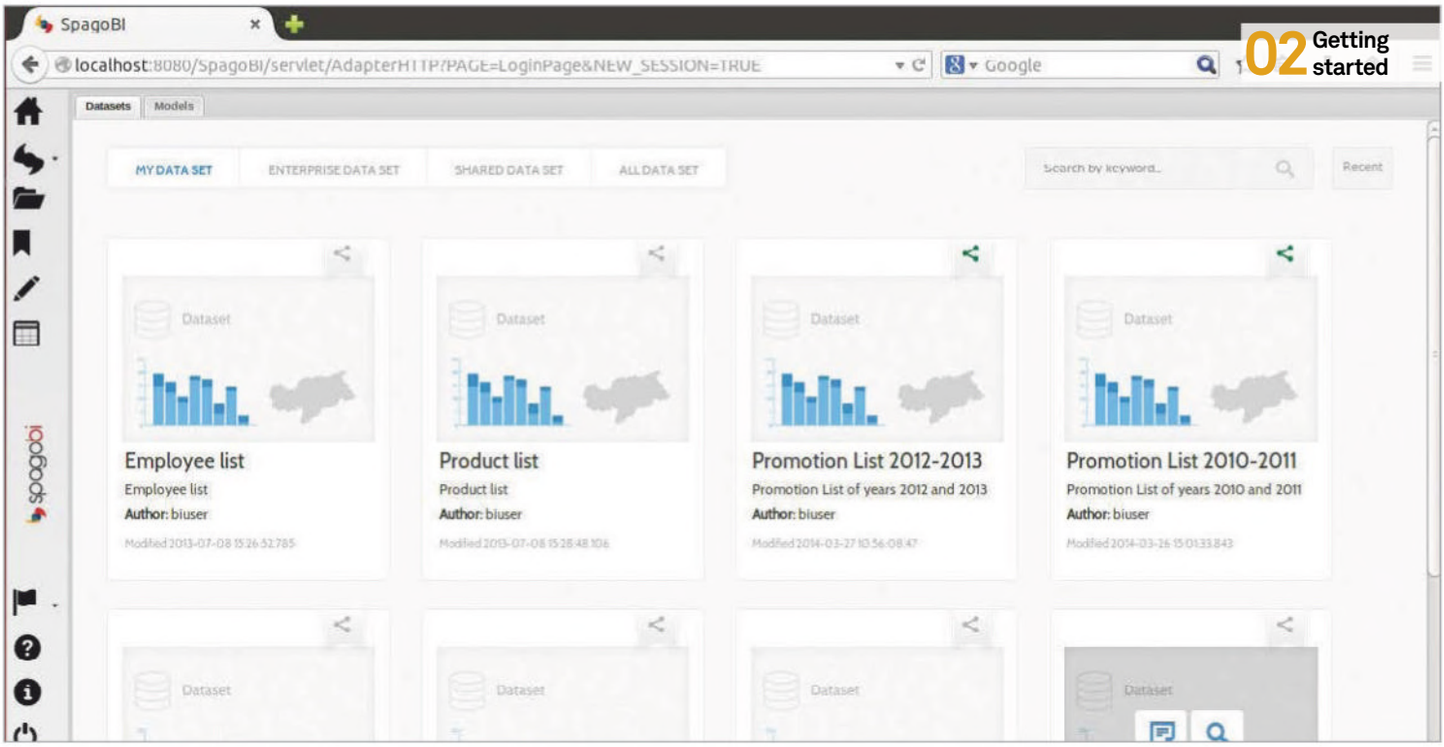
Nitish Tiwari is a software developer by profession and an open source enthusiast by heart. As well as writing for leading open source magazines, he helps firms set up and use open source software for their business needs

Resources

SpagoBI home page
www.spagobi.org

Any business exists for one sole reason – to get customers. However, the business environment today makes sure companies are constantly on their toes, as a simple mistake can quickly take customers away. Business intelligence (BI) is the field that aims to diffuse this situation. Business intelligence can be defined as a set of tools and techniques for getting meaningful and useful information out of raw data in order to help better analyse the business in question. Or more simply, business intelligence tools let you decide what's right and wrong for your business, based on the data from it.

You may think 'if the data is from my business, why can't I just make decisions right away, instead of using software tools in between?' The reason lies in human evolution: we evolved in an environment where quick decisions needed to be taken based on visual information, so we are generally bad with numbers, especially when dealing with lots of them. But today we'll introduce you to the open source BI tool SpagoBI, which does the hard work for you. You will see the installation steps and then learn how to make the best decisions with SpagoBI on your side. For demo purposes, Ubuntu 14.04 has been used as the host system and SpagoBI stable version 5.0.0.



01 Begin the Installation

To begin the installation, you need to download 'All-In-One-SpagoBI-x.x-yy.zip' from **http://forge.ow2.org/project/showfiles.php?group_id=204** (x is the version and yy is the release date). The zip file has the SpagoBI server and database rolled together, so there is no need to download anything else. Note that SpagoBI uses the default Tomcat port 8080, and you may have problems if you have previously installed Jenkins or other software that uses Tomcat port 8080. It's better to start with a fresh Linux installation and you also need Java installed on your computer. To check if its already installed, execute the command **java -version** in the terminal; if you get a response with that command, you're good to go. If the command returns something like 'The program java can be found in the following packages', then install java using **sudo apt-get install default-jre** (on Ubuntu). Once you have done this, continue the process.

After downloading the file, unzip it and go to the bin folder via the terminal. Type the command:

```
cd /<location of download>/SpagoBI-server-5.0/bin.
```

Grant permission for the shell scripts to be executed and use the command **chmod755*.sh**. Also, navigate to the /database folder and change the permissions in that folder too. After changing the permissions, return to the bin folder. Note that by using '*.sh' we are changing the permissions for all the files with extension .sh – not a good practice in production environments.

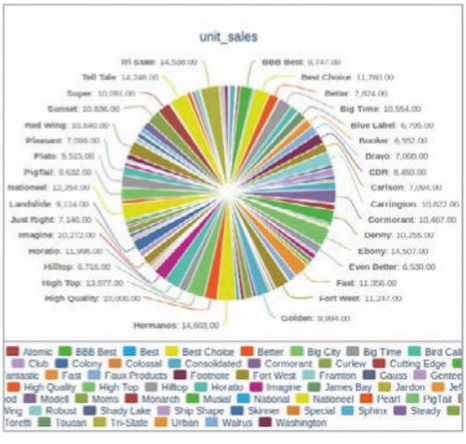
Execute the shell script to run SpagoBI. Use the command **./SpagoBIStartup.sh**. If you see a message like 'Start up sequence completed in xx ms', go to your browser and access SpagoBI using the URL **http://localhost:8080/SpagoBI/**.

Once the page opens, you can log in using the default credentials mentioned there.

02 Get started

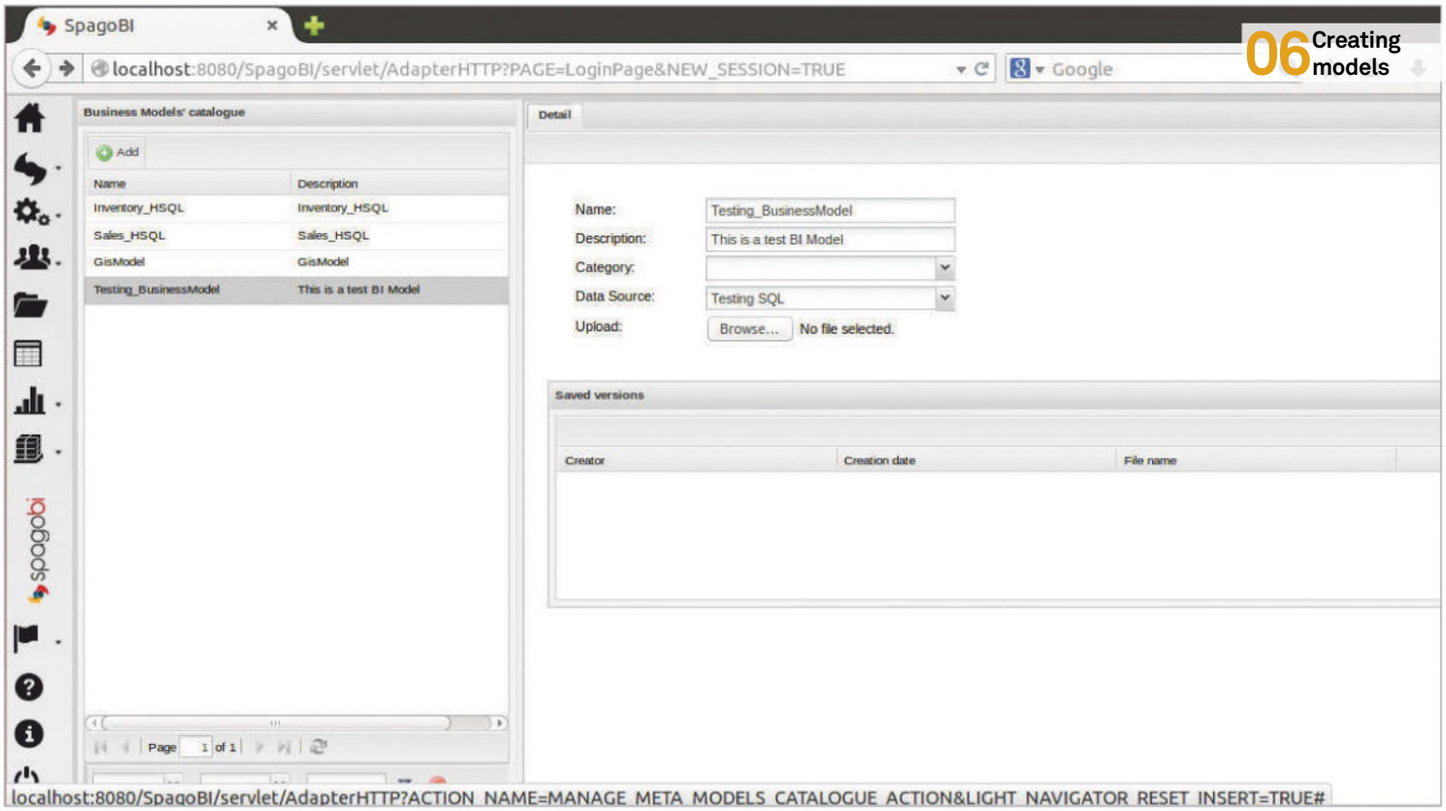
As a BI tool, SpagoBI's main focus is to help generate actionable analytics. To achieve this, SpagoBI connects with a range of data sources (like databases) and provides a simple and easy-to-use GUI to help you create analyses. If you just want to explore the features, go ahead with the default credentials mentioned on the log in page. In a production environment though, you will need to create users. For demo purposes, the biuser and the biadmin users have been included.

Now start creating your analysis. Log in as biuser and click on the My Analysis link. In the new page that opens, click on Create Analysis in the top-right corner. The next page will ask for the reporting type; select 'ad hoc reporting'. You can now select the appropriate dataset for your analysis. Since there is no dataset created yet, use the sample one.



03 Create the analysis

Once you hover on a dataset, you get two options – 'show worksheet' and 'show QbE'. Select the worksheet option to open up the worksheet designer (QbE will be covered later). The designer lets you select the visualisation style, like a bar chart, pie chart, line chart and different tables. Once you select the style, drag and drop the data fields to the designer. Note that these fields are based on the dataset you first selected, and they serve as the input to the graph. After the data is fed, preview the analysis using the Preview button in the top-right corner. If everything looks right, click the Save button above Preview. After an analysis is saved, you can view it later under the My Analysis menu.

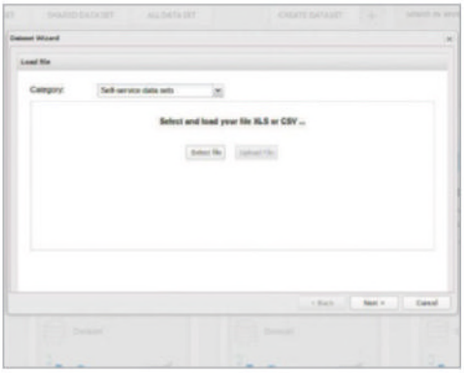


04 Define a query with QbE

The ability to draw graphs from a fixed dataset isn't enough for real-world scenarios. You may want to look at things from multiple perspectives. With QbE you can define your own query graphically, execute it, check the results, export them, save the query for future use and generate a reporting template. To start, select QbE during the analysis creation. A schema window (related to the selected datasheet) and a query editor will open. To create a new query, drag and drop the relevant dataset fields to the query editor on the right – you can apply filters. Once the fields are populated, query is generated automatically. View, using preview, or create graphs for this new query using the worksheet tab. QbE also works on data models that are made available by the admin.

05 Create datasets

As in the previous steps, datasets form the base for creating analysis, so now see how to



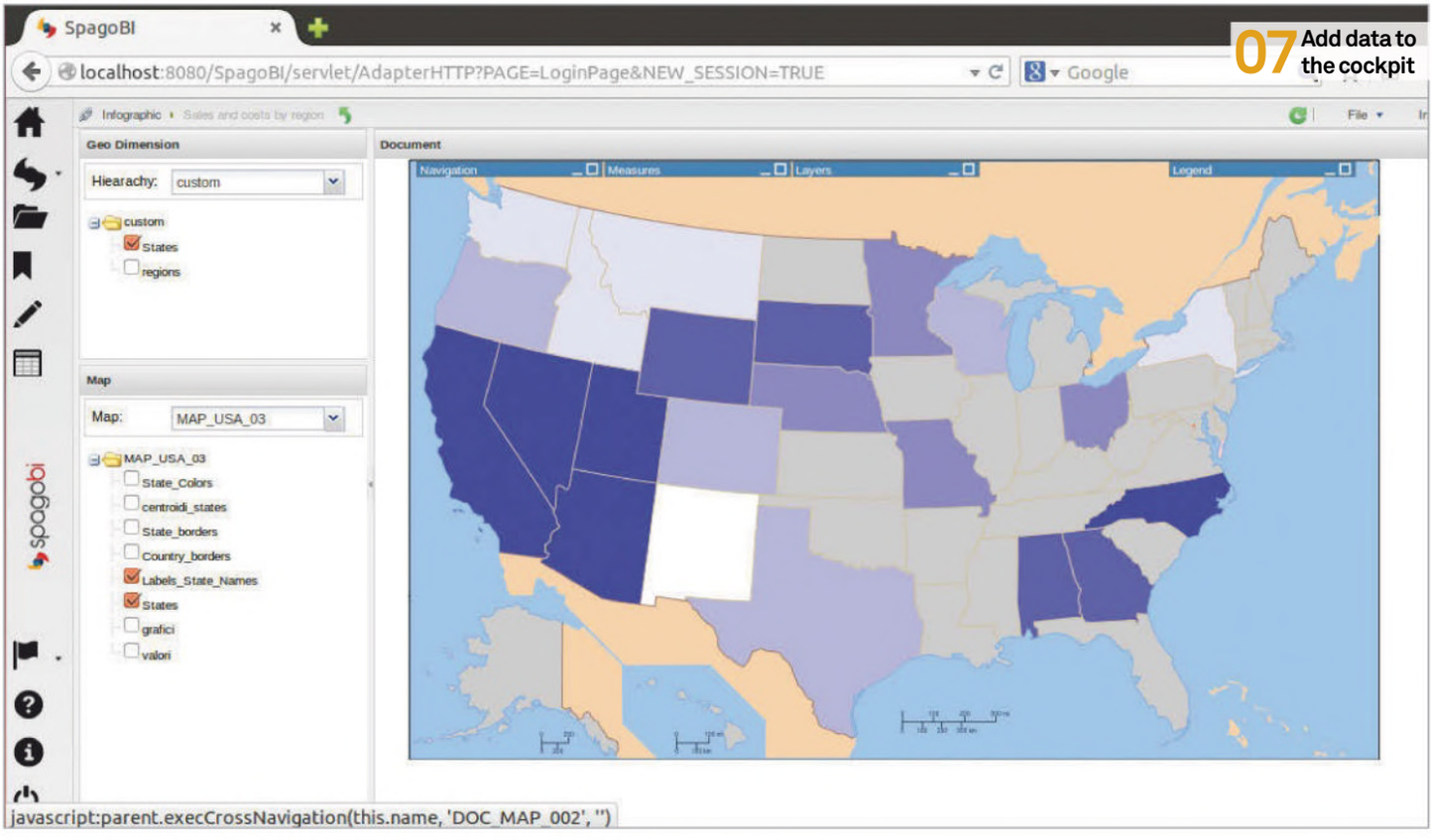
create your own datasets in SpagoBI. Click on the My Data link on the left menu bar and you will see all the preloaded datasets. Then click on the Create Dataset button in the top-right corner of the page. The new pop-up asks you to upload an .xls or .csv file containing the data. Once you have uploaded the file, you need to select the delimiters, quote characters and encoding of the file. SpagoBI will automatically identify the column headers. Then define the attributes and values for each of the columns. Once done, the data is ready for preview, so check if everything is the way you want it to be before saving. The dataset is now available for you to play with.

"SpagoBI will automatically identify the column headers. Then define the attributes and values"

06 Make models

The main difference between models and datasets is that while datasets are predefined data imported from different sources (like Excel files), models are related to the databases directly. You can create datasets from models too. Models, however, can be created by admins only. To create a new data model, you need to log in as admin first. First, see how to add a data source like a new database. Click on the resources link and then go to Data providers>Data Source. Here, you will see two preloaded databases. Click on the Add New button and a new form will open on the left – fill it with the relevant details (database connection specific details). Once filled, click on the Test link in the top-right corner to check if SpagoBI can connect to the database, then save it.

Click on the resources link and go to Catalogues> Business Models Catalogue. Here, add a new model and select the data source as the newly added database. Save the model to finish the process. You can now see the new model under the My Data link.



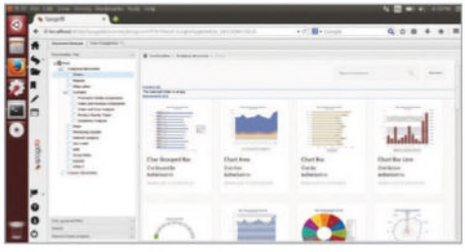
07 Add data to the cockpit

Cockpits provide an interactive way to enable faster data insights. With data mash-up support, you can add enterprise data or externally sourced data to your cockpit. To create a cockpit, go to My Analysis> Create Analysis>Cockpits. In the page that follows, you have a clean canvas available. Add widgets to the canvas using the Add Widget button in the top-right corner. After clicking the button, a new blank widget will appear. You need to then configure the widget with the data you'd like to be displayed. The widget setup is roughly the same as the analysis setup in Step 3. You can add as many widgets you please, and then save it for future viewing.

08 GEO and GIS engine

SpagoBI GEO engine enables users to re-aggregate information dynamically, according to a geographic hierarchy (nation, region or district, for example) defined by the administrator. This engine can be used irrespective of geographic context, so you can display the distribution of indicators on any structure that can be represented on a map, including process flow diagrams and hardware infrastructure topology. To create a geographical analysis, go to My Analysis>Create Analysis>Location. This interface lets you select the hierarchy, the level of integration (with extra charts and values) and the map selector. You can also zoom in and out of the map, access legends and do most things possible with other graphs.

SpagoBI also has the GIS engine, which helps the visualisation of business data on a map. The trick here is that you can select the cartographic layers that you'd like to see the data on. For example, you may think viewing the sales data state-wise is cool, but what if you could view the sales data based on population concentration? Wouldn't it provide a better idea of the sales quality? Take a look at SpagoBI GIS engine to see this in action.



09 Access the document browser

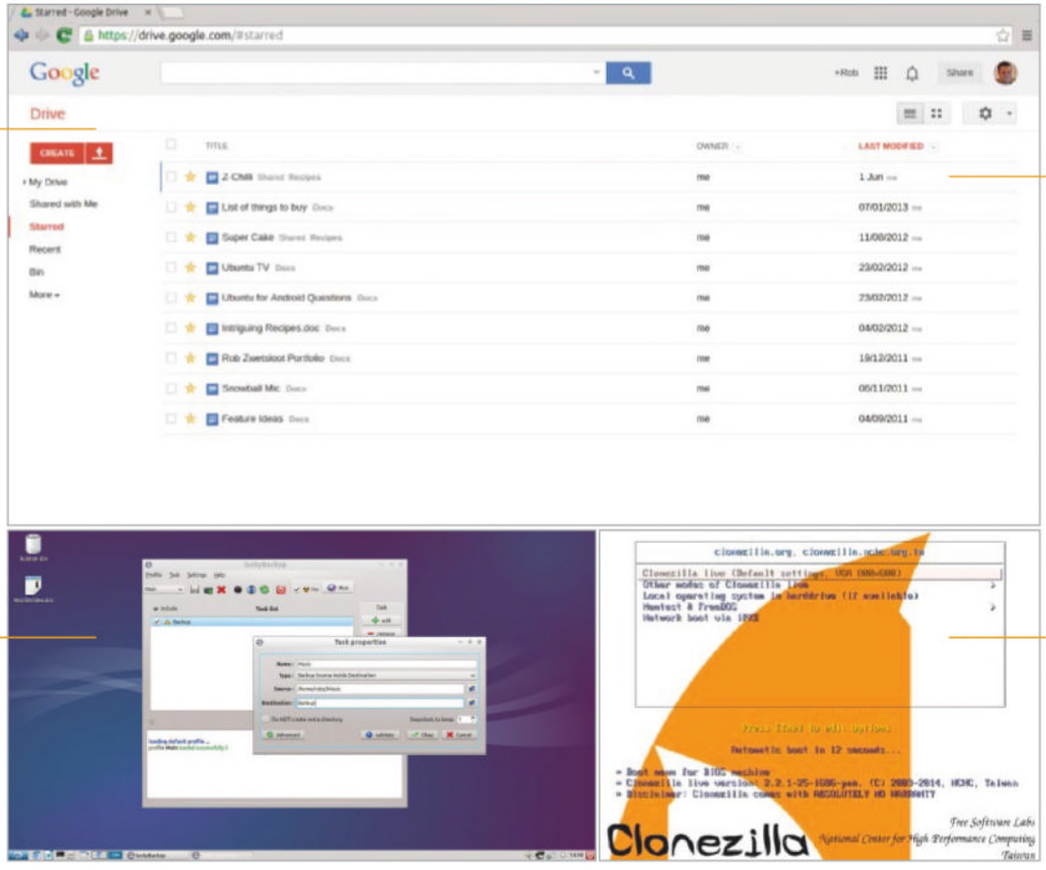
The document browser gives access to the functionality tree containing all SpagoBI analysis documents and folders. Wherever you may have saved your analysis, you can find it here with the document browser. There is also the search, sort and filter functionality available to help you handle the documents properly. Note that the documents available in the browser are clickable, enabling you to get the data directly once you find the relevant document. To get started, click on the My Documents link in the left menu bar.

10 To finish

While SpagoBI is the first 100% open source BI tool, it is also huge in size. It has so many features that it would probably take a full magazine to describe all of them in detail. Combined with the complex field of data analytics, you would probably get lost in the details, especially if you are a beginner. However, as they say, the journey of a thousand miles starts with the first step, so don't be intimidated but consider this article and SpagoBI as the stepping stone to get you started on your journey of data analytics. Not only will this help you understand customer behaviour, but you will also get some great insights into data visualisation more generally.

Use a variety of popular cloud services to properly back up your system

By performing occassional backups, you create a grandfather system



Schedule file backups so you can quickly and easily save important files to the cloud

Create perfect copies of your system with Clonezilla for a thorough backup

Back up to the cloud

Advisor

Rob Zwetsloot models complex systems and is a web developer proficient in Python, Django and PHP. He loves to experiment with computing

Resources

Cloud storage account

luckyBackup luckybackup.sourceforge.net

Clonezilla clonezilla.org

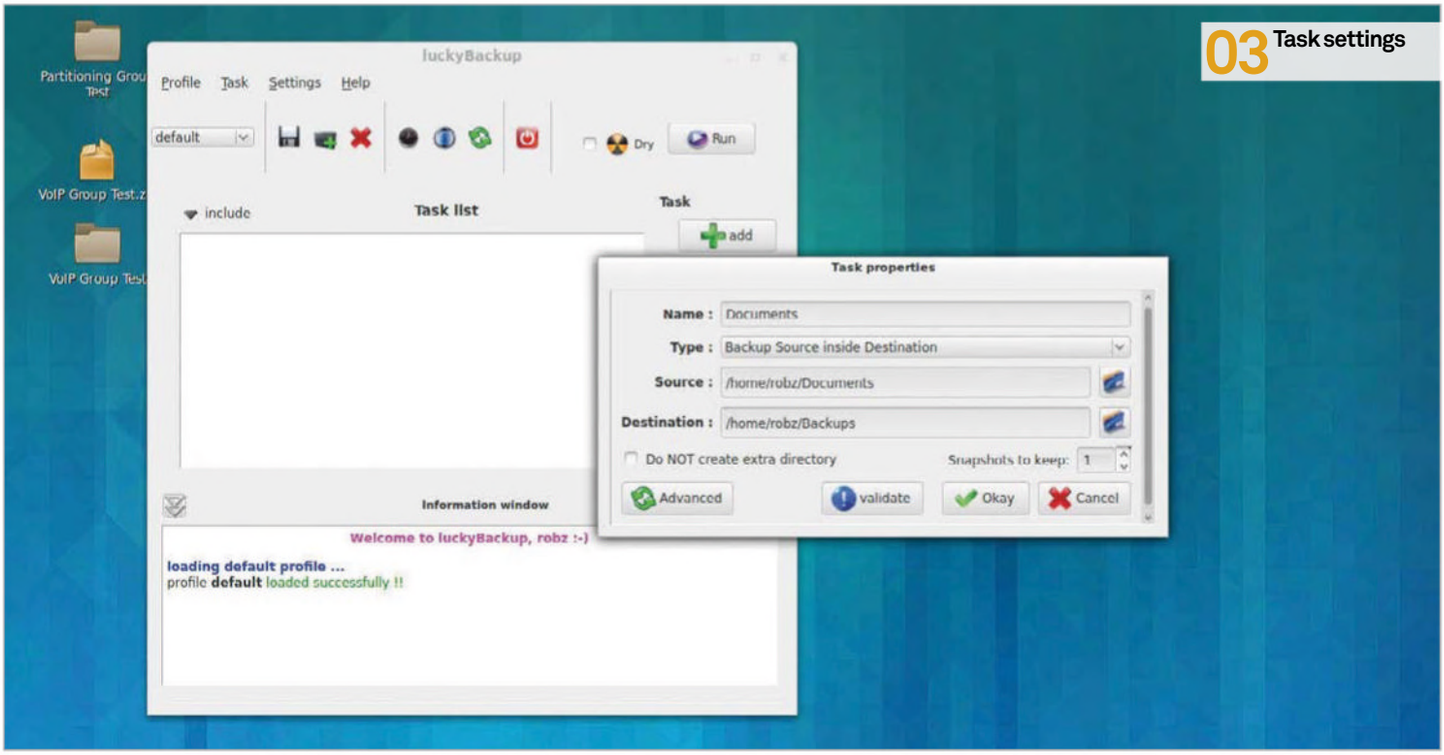
Automatically back up your files or entire system and send directly to the cloud

Backing up is always important, but one of the most important aspects of backing up is the storage of said backup. There are various levels of security you can give your backups; you could keep them on the same computer in case something goes wrong with the original files, for example. Alternatively, backing up to a different system in the local network means your files are safe if there's a hard drive or other catastrophic failure.

An offsite backup is still the safest option though, protecting against even greater threats like fire or theft. Truly, the best way to back these files up with the smallest risk of losing them is to send them to the cloud.

Large cloud storage solutions have the advantage of keeping data safe even in the unfortunate event that a data centre has gone down, which means the possiblity of losing your backed-up data is very low in this case.

In this tutorial we'll show you how to properly back up files and certain aspects of your PC, and then show you how these can be sent to the cloud service of your choice.



01 Grab luckyBackup

Install luckyBackup, a great little backing-up application that contains scheduling features, customisable backup tasks and profiles for grouping tasks. Install it from your repository; the package name is **luckybackup**.

02 Set up profiles

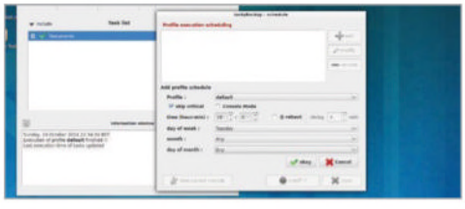
Back up your important documents – we'll assume you're keeping them in Documents, in your home directory, but if they're placed anywhere else then just switch out that folder for Documents. Go to Add underneath Task to begin.

03 Task settings

Name the task whatever you wish so you can remember what it is, choose the Documents folder as the source and, for now, create a new folder called 'backup' for all the backed-up files. Keep the Type field the same. The benefit of doing the backup this way is that only new or updated files get included.

04 Test task

Before we go any further, it's best to test the task you've created. Click the check box for the task and it will display a triangle to let you know that you haven't done a backup before. Click Run to perform the first backup of these files and it will let you know if there are any errors.

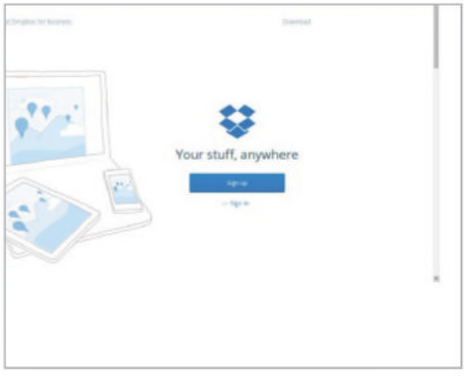


05 First scheduling

On the main menu, click on the clock symbol next to the red x to bring up the scheduling window. Here you can select when backup profiles are performed, down to the hour, day and even month. These are done on a per-profile basis and can also be activated on a reboot.

06 Extra profiles

The scheduling is performed per profile, so if you have different files you want to back up at different times then you'll have to create separate profiles. You can group multiple tasks under one profile if you need to back up multiple locations at once by using Profile>New.

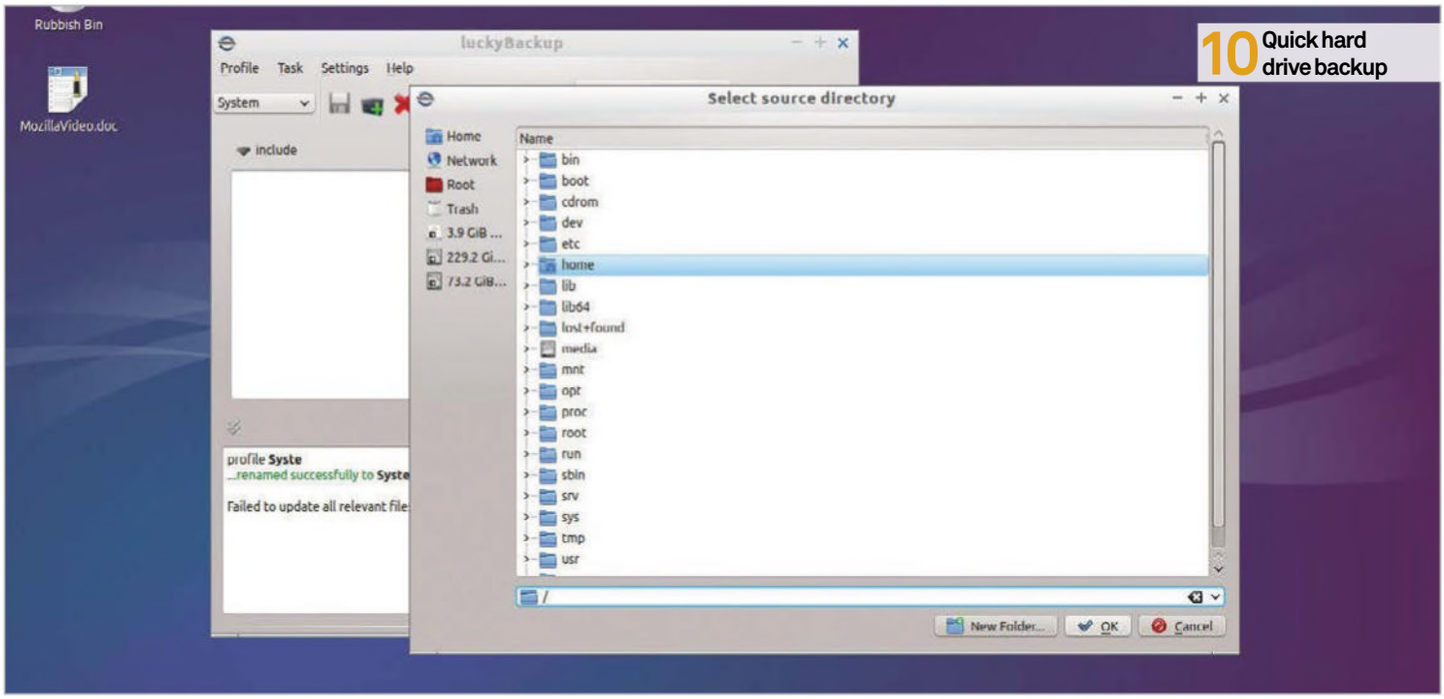


07 Choose a cloud

Any cloud service will be fine for our purposes, however our main concern is space. Documents won't take up much space at all – it's less than you'd think – but once you get to music, video and the disc images we plan to upload later then the required space begins to increase rapidly. Choose your service wisely.

08 Connecting to the cloud

Cloud services that work properly on Linux, such as Dropbox, will create a folder in the home directory for syncing files or let you choose a folder to sync. If you're using your cloud space for other files, we suggest creating a backup folder inside the sync folder for you to work with.



10 Quick hard drive backup

09 Multiple computers

One of the benefits of syncing all the files to Dropbox is that they can also be downloaded to another system. You can either do this to create a more accessible backup, or back up multiple sets of documents to the cloud. Make sure any files that might clash are kept separate.

10 Quick hard drive backup

The full root system of your computer can be backed up quickly. It will save all your files and programs without needing to make a big disc image. This won't be a complete backup as such though, as it won't remember permissions very well, but it *is* better than nothing. When creating a backup task set the source as /.

11 Backup location

You can't set the location to be the original hard drive as this can cause errors when trying to copy what you're creating. Either use a separate or external hard drive that's large enough to contain the files or have a spare partition purely for the hard drive backup.

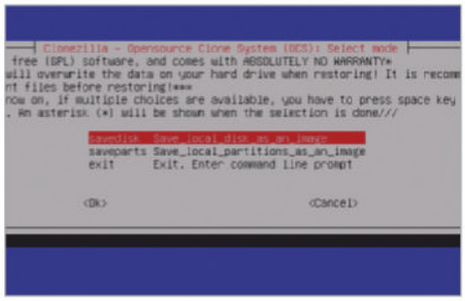
"The best software for this is Clonezilla"

12 Complete image backup

You won't be able to do this within the running operating system. The best software we can suggest for this is Clonezilla, which is a live disc that runs ghosting software. It can be obtained from the Clonezilla website (clonezilla.org).

13 Use Clonezilla

Write the Clonezilla ISO to disc, reboot your system and make sure you boot from disc. Follow along with the menus to select your language and resolution until you get to the first proper Clonezilla option screen – choose device-image, as we want to create an image from a device.



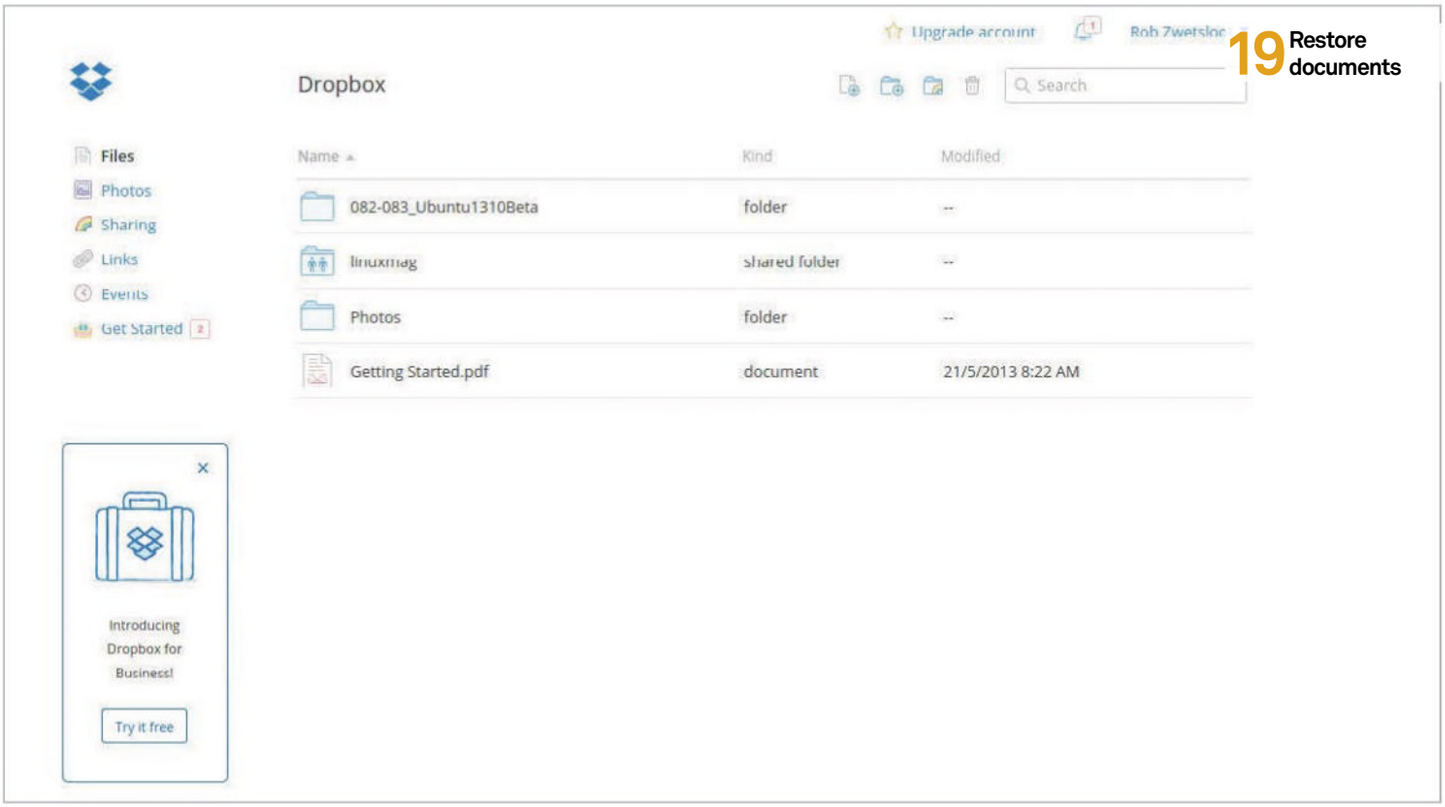
14 Create the image

Choose local_dev so you can select a local hard drive or partition, and then select the partition or drive where you want to save the image to. Choose beginner mode, and then choose whether you want to save the entire hard drive or just partitions from the hard drive. Finally, select the partition, and then go through the menus before finally hitting Yes to start.

15 Upload concerns

The resulting image will be large, easily totalling in the tens or hundreds of gigabytes depending on what exactly goes into the backup. Uploading this will take a while, and some cloud services have limits on the size of files you can backup.

It also won't just change the differences in the image and will entirely replace it each time. We suggest doing this kind of backup less often – once every week or month depending on your needs and data allowance.



19 Restore documents

16 Break down images

One method you could use to try and make uploading easier is to split it up into multiple zip files – this won't compress the image, but it will make the files much more manageable to upload to the cloud. In Linux we can do this in the terminal by first turning it into a zip file with:

```
$ zip image.ISO
```

17 Split the zip

Once the zip has been created (it may take some time), go back to the command line. Decide what size you want the chunks to be – 100MB usually works well – and then split them with the following command:

```
$ split -b 100M image.zip
```

18 Bring the zip back together

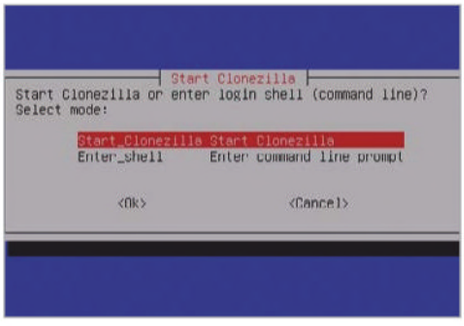
It's easier to split the files than it is to put them back together. Once you've downloaded all the necessary parts, you'll need to make sure they're all named similarly (something like **image1.zip** for example) and then you can bring them together with:

```
$ cat image* > ~/backupimage.zip
```

"You can group multiple tasks under one profile if need be"

19 Restore documents

Restoring your documents is extremely easy – you'll just need to download them from the cloud backup and put them back in their original place. Services like Dropbox allow you to choose previous versions of a file, in case any newer ones are corrupted as well.



20 Restore an image

Download the image and create another Clonezilla live disc or USB if you need to. Have the image attached in some way to the system and go back into Clonezilla. Again, we want to go to device-image as we will be restoring from an image.

21 Choose the image

Use the options from before, being sure to choose the hard drive you wish to restore to as the destination. On the screen where you would usually hit savedisk or savepart, look for the restore disk option. Choose the image and the hard drive to restore to again and begin.

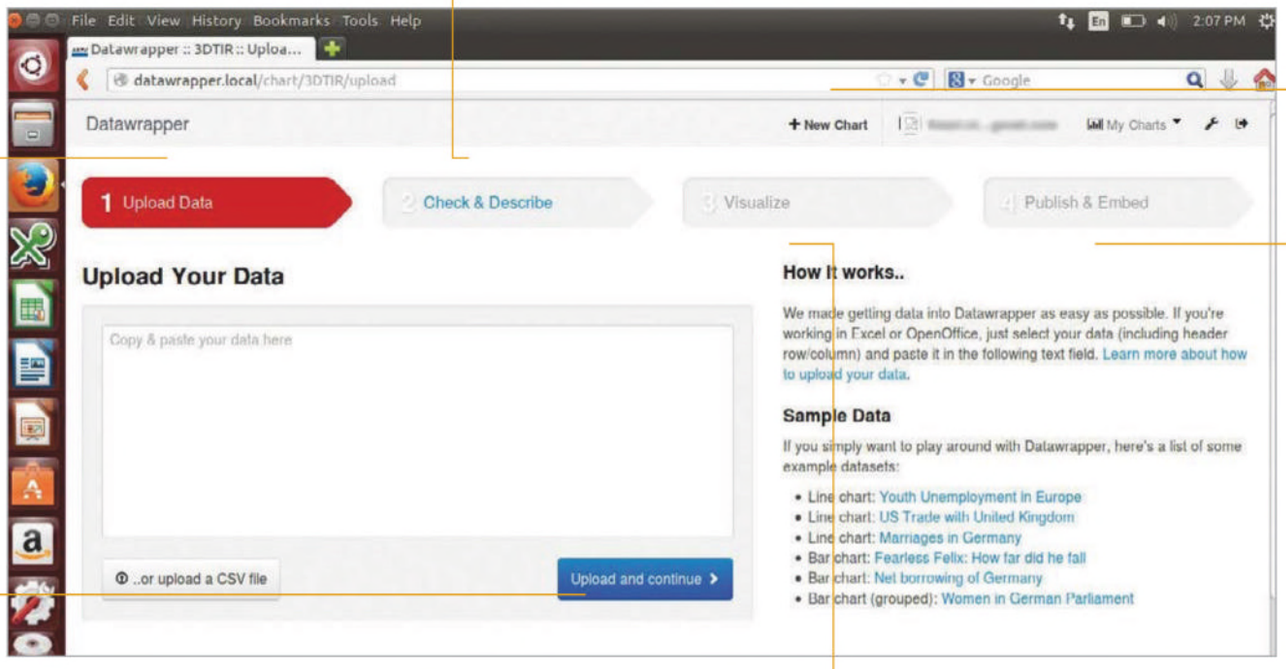
22 Automation for disks

Unfortunately, there's no way to automate the disk imaging process unless you create scripts and do it via virtualisation. However, it shouldn't be necessary to do this kind of imaging on a regular basis. With a bit of practice and playing around with settings on everything, though, you should be able to make the backing up of documents and images an easy process that takes very little maintaining.

This section lets you upload your data. It serves as the starting point to create new charts

Here you can check the data format, uploaded in the previous section. You can also edit the data

New chart button leads you to a blank page for creating new charts. You can also access all the saved charts here



Once the data is pasted in the text area, click on the "Upload & continue" button to move to the next step

Select the chart type to visualise the data. Once the chart is selected you can also set parameters for the charts here.

You can choose to get the code here and embed it directly to another webpage or export an image of the chart for printing

Visualise your data with Datawrapper

With average attention spans falling, data visualisation is a very important way to put your point across quickly and efficiently

Advisor

Nitish Tiwari is a software developer by profession and an open source enthusiast by heart. As well as writing for leading open source magazines, he helps firms set up and use open source software for their business needs

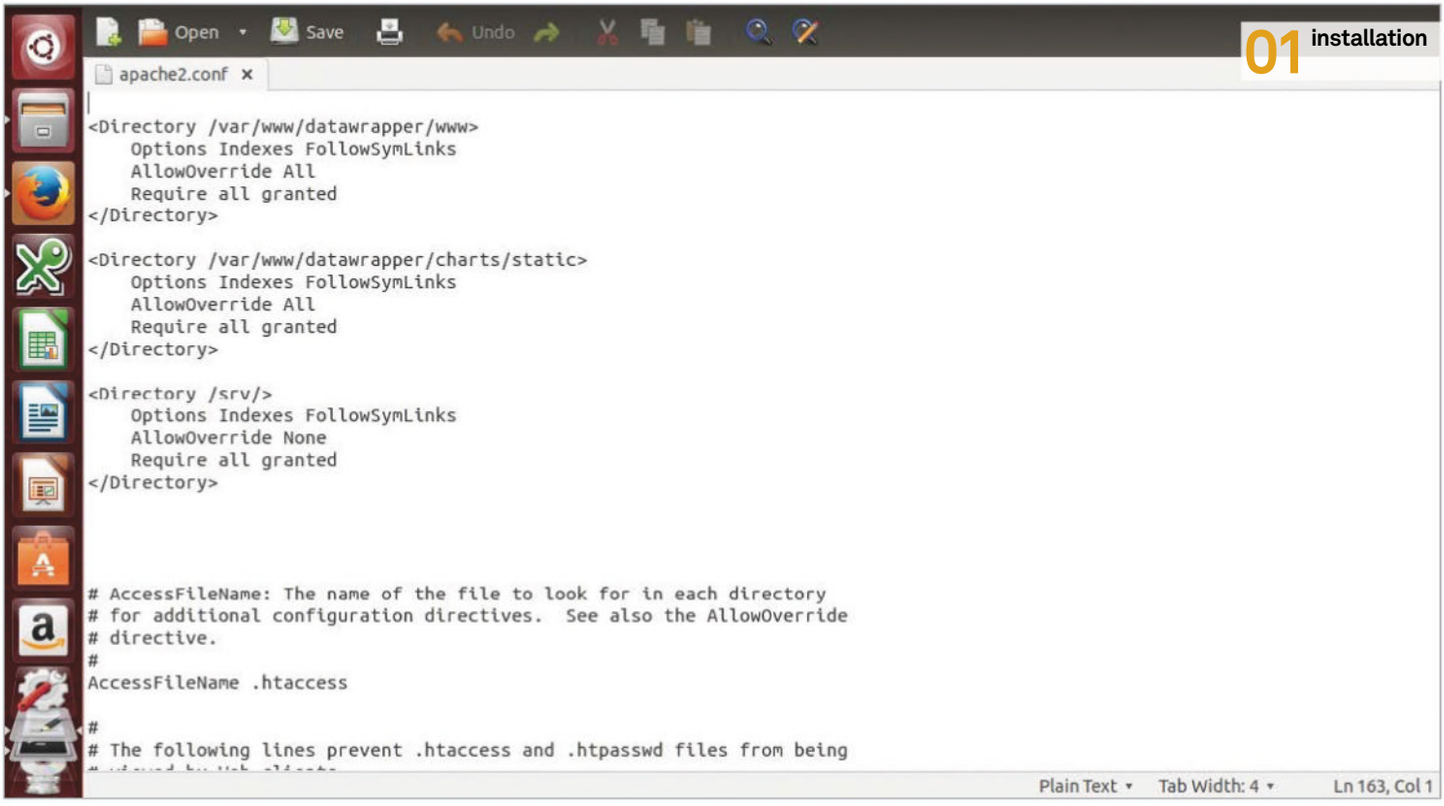
Resources

Datawrapper home page

www.datawrapper.de

A recent study from the National Center for Biotechnology Information found that the average adult's attention span has now dropped to a mere eight seconds from 12 seconds a few years back. While this drop doesn't seem to be too much from the previous statistic, it means that you potentially have even less time to put your point across. It is very important for us to now communicate in easy-to-understand yet catchy language. How can you create compelling stories every time? The answer may lie in the common phrase 'a picture is worth a thousand words'; our minds can process images around 60,000 times faster than text.

So why not convey key messages with images and graphs? In this tutorial, we will introduce you to Datawrapper, which helps you to convert boring, raw data into easily comprehensible graphs. It's based on the server-client model and you can install Datawrapper onto a server – anyone on the network can then access it via their browser. After creating a graph, you can either embed it in a webpage (the server should be accessible from the webpage) or export it as an image for printing. With support for several kinds of graph, Datawrapper has you covered for all data types. We will now start with the installation process, and then move on to plot graphs.



01 Installation

Before we start with Datawrapper installation, you need to have Apache server (with mod_rewrite and .htaccess enabled), PHP (version 5.2 or above) and MySQL server installed. These are all available under one package – the LAMP server. To install the LAMP server, first refresh the package index using **sudo apt-get update**. Then install the package using **sudo apt-get install lamp-server^**. To enable mod_rewrite use the command **a2enmod rewrite** and then restart apache using **service apache2 restart**. You can also install phpMyAdmin (optionally) if you are not comfortable using the command prompt for databases. To install phpMyAdmin use **sudo apt-get install phpmyadmin**.

Due to its design, a separate domain is required for datawrapper. It's not possible to run it in a subdirectory, ie **http://localhost/datawrapper** will not work. Let's now get started with the installation.

Download and unzip the Datawrapper repository. It is available at **https://github.com/datawrapper/datawrapper**. Navigate to the datawrapper folder in command prompt and type: **curl –sS https://getcomposer.org/installer | php**. This installs composer onto your server. Now run **php composer.phar install**. This downloads all the dependencies required by Datawrapper. Create a new MySQL database and initialise the table schema using **/lib/core/build/sql/schema.sql**. You can simply import the schema file if you are using phpmyadmin. After the database is created, rename the file /lib/core/build/conf/datawrapper-conf.php.master to datawrapper-conf.php and update the dbname, dbuser and password in the file.

We have used Ubuntu 14.04 as the host system and Datawrapper version 1.7.11 for installation.

02 Web server configuration

Now that the dependencies are installed and the database is ready, vwe need to create two new virtual hosts – one for the Datawrapper instance, eg **http://datawrapper.local** (pointing to /www folder inside the datawrapper folder) and another for datawrapper charts, eg **http://chart.datawrapper.local** (pointing to the /charts/static inside the datawrapper folder). First create two copies of the default configuration file (000-default.conf), and name them as per the host names. Use the commands:

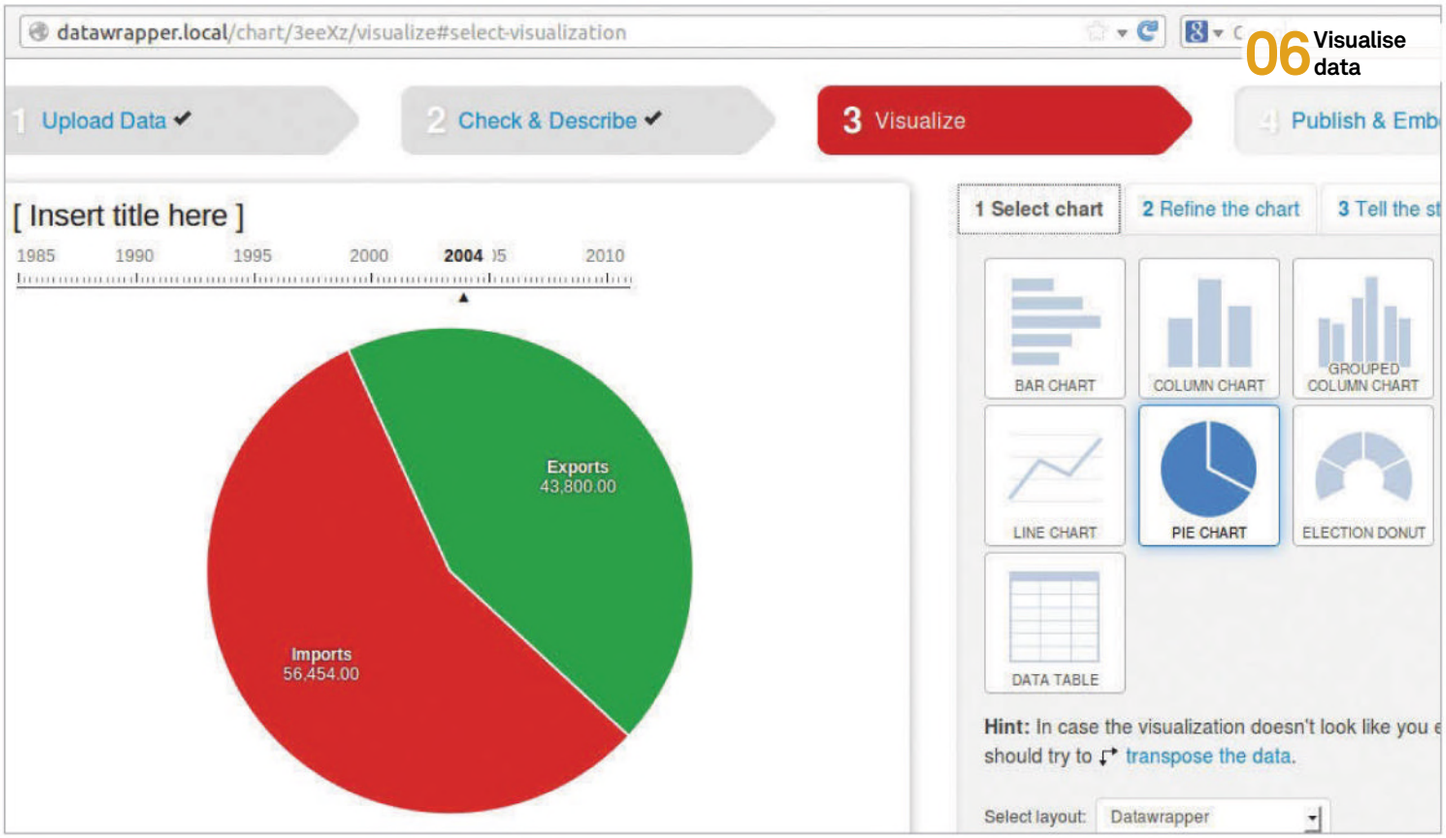
```
#sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/datawrapper.local.conf

#sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/chart.datawrapper.local.conf
```

Then update the DocumentRoot and ServerName fields in both the files created above. Enable the access to the document root added above in the apache2 configuration file (available at /etc/apache2/apache2.conf). This is required because apache2 is generally configured to not allow access to root file system outside /var/www. Enable the sites by using the command:

```
#sudo a2ensite datawrapper.local.conf

#sudo a2ensite chart.datawrapper.local.conf
```

Then you need to create the configuration file by copying the config.yaml.template file to config.yaml. Update the domain, chart domain and the email address in the file. Also, make sure the /charts folder (and everything inside it) is writable by the web server process. Then install the core plugins using **php scripts/plugin.php install "*"** and run **make** to build the JavaScript library.

This completes the installation process; you can now access datawrapper at **http://datawrapper.local**, and if everything is fine then you'll see the message "Congratulations! You have successfully installed Datawrapper".



03 Get started

Before creating your first graph, you need to create an account to be able to embed your charts to other websites; there is little use in creating and keeping the graphs yourself. Click on the "Login/ Sign Up" button at the top-right corner to create your account, then log in using your credentials. Next step is to activate the email id you entered while creating the account. If your Datawrapper host has the email server configured, you will get an email with the validation link – click on the link and you are ready to go.

If the email server can't be activated for some reason, or if you have installed Datawrapper on your home PC just to have a look at it, you need to validate the email manually. To do this, go to the phpmyadmin on your host and open the user table inside Datawrapper's database. Go to the column activate_ token and copy the string (corresponding to the email you want to activate). Now, go to your browser and access the URL **http://datawrapper.local/account/ activate/<activate_token>**. (Substitute <activate_ token> with the string copied from database.)

04 Data upload

To create a chart, click the "New Chart" button at the top-right corner. On the next page, you need to upload the data – anything that you would like to plot in the form of chart. It just needs to have at least a pair of data points. Adding data to Datawrapper is very easy as well. If you are working in OpenOffice or Excel, simply copy and paste the data (including row/column headers) into the text field marked "upload your data". If you have a CSV file, you can directly upload it. If you don't have data at the moment but plan to look at how things work, there are a few sample data sets available under Sample Data, so just click on the link you want to use and the data gets populated. After uploading the data, click on the "Upload & continue" button.

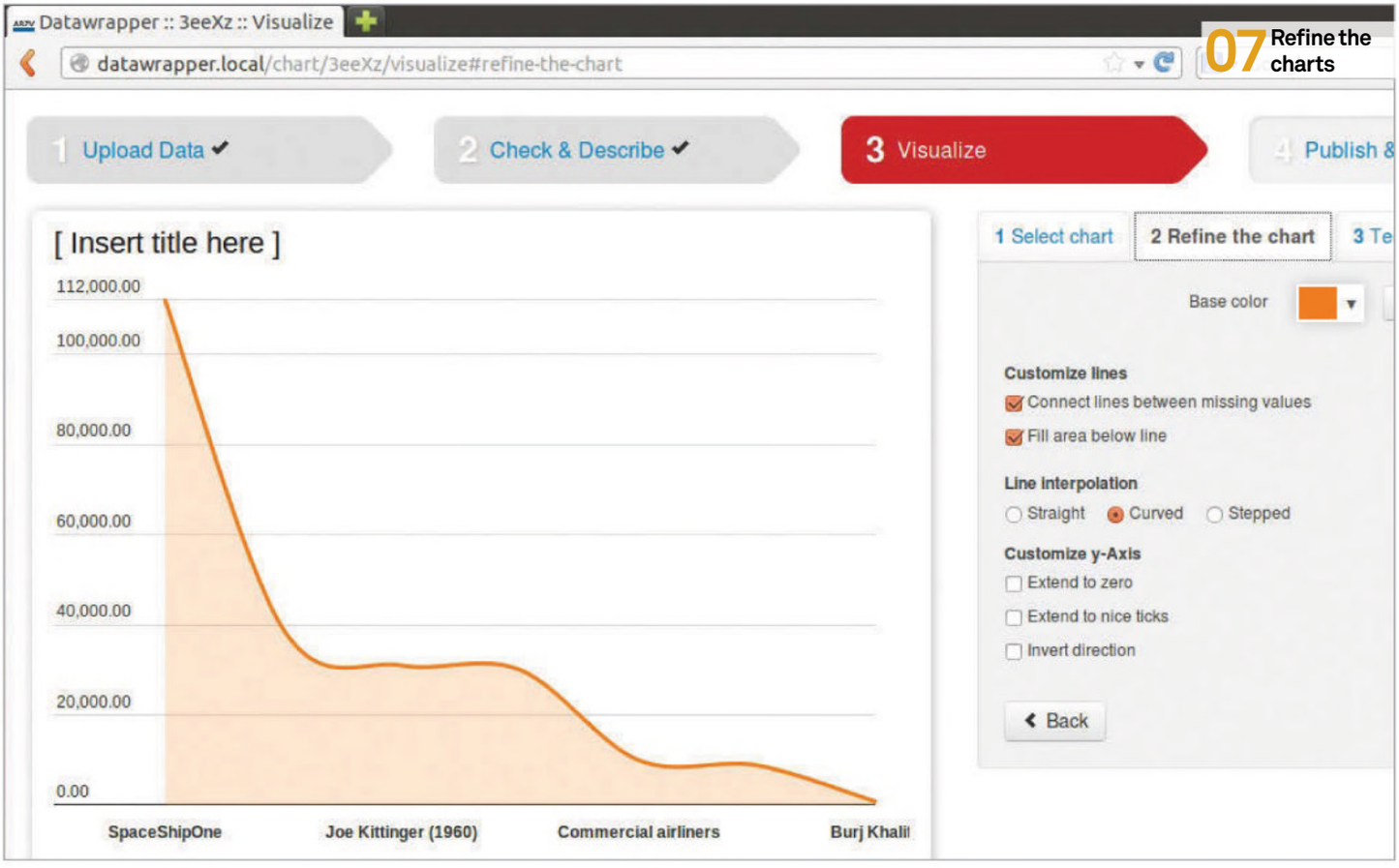
05 Check and describe data

The next step is to check and describe the data you just entered. As you paste/upload the data, Datawrapper automatically checks and displays it in a tabular form. You need to carefully inspect it to make sure that the data is interpreted in the way you want it. If there is a change required, you can edit the table directly. You can control whether the first row is interpreted as label or data. You can also credit the data source; it will be shown in the bottom-right corner of the map if you update the fields under Credit the source. There are a couple of other options: to customise the columns use the check box on the column header. As you click, a new menu opens up on the right side. Click on the row-column intersection to transpose the table. After the validation is done, click on "Visualize" to go to the next step.

"The next step is to check and describe the data you just entered"

06 Visualise data – various chart types

Data can take any shape or size, so charts need to be flexible too. The visualisation tab lets you simply click a chart type to visualise the data entered in the previous step. You can select bar chart, line chart and a few other types of column and pie charts to visualise your data. With smart data visualisation, you don't really need to bother with the nitty-gritty of plotting charts – if the chart doesn't look as you imagined it would, just click "transpose the data" link and the chart will take a comprehensible shape. If there is still a problem, you can select a different chart type.



"The "Tell the story" tab lets you edit the title of the chart"

07 Refine the charts

After you select a chart type for your data, it's time to fine-tune it to suit you. Click on the "Refine the chart" tab; you can find options based on the chart you selected earlier here.

- Bar/column charts: If you have selected, bar/column charts, you get the option to customise the base colour. You can also choose to automatically sort the bars and reverse their order.
- Line charts: In addition to changing their colour, you can choose to fill the area below the chart, set the interpolation to straight, curved or stepped. You can also customise the y axis here.
- Data tables: You can add sorting to data tables using the refining option. The table can also be displayed in several pages, if it has several columns.
- Pie/donut charts: You can edit the colours for these types of charts.

08 Add story to the chart

Now that the charts are customised, let's add story to the chart. The "Tell the story" tab lets you edit the title of the chart and add some description to the chart itself. This info gets displayed at the top-left corner of the chart window. You can highlight the important elements of the chart, using the drop-down available below the description window. If you forgot to credit the chart data source in the "Check & Describe" section, you can do that in here as well.

09 Publish & Embed

This is the final step in the chart creation process. You can view your chart in full glory here. If you have already validated your email id – as discussed in Step 3, you will also get a ready-to-use code snippet under the "Embed into your website" section. You can just copy and paste it to another webpage, and the chart is shown there – given that your server is accessible from the webpage. For example, if you have installed Datawrapper on a computer in your local network, you can display the charts within your network. Below this, you also have the option to export the chart to a static image that can be used for publishing etc. Towards the top, there is a direct link to the chart available as well.

10 Finishing thoughts

Datawrapper is a great tool that enables almost anyone to create beautiful charts – there is no need for any sort of technical or mathematical knowledge. If you have some formatted data, Datawrapper can almost certainly plot it. While the installation process is a little lengthy, it is a breeze if you just follow these few steps carefully. If you have visited the Datawrapper website, you'll be aware that there are few new features available now as well. Prominent among them is the choropleth map feature, which lets you show data in a geographical map format. Though this is not currently available in the GitHub source, we hope it will be soon.

Get your smart TV in on the action by connecting your HTPC to it with a HDMI cable

Build a Linux HTPC

Set up the most powerful home theatre PC possible using a custom Linux setup, with tips on hardware and software

Advisor

Rob Zwetsloot models complex systems and is a web developer proficient in Python, Django and PHP. He loves to experiment with computing

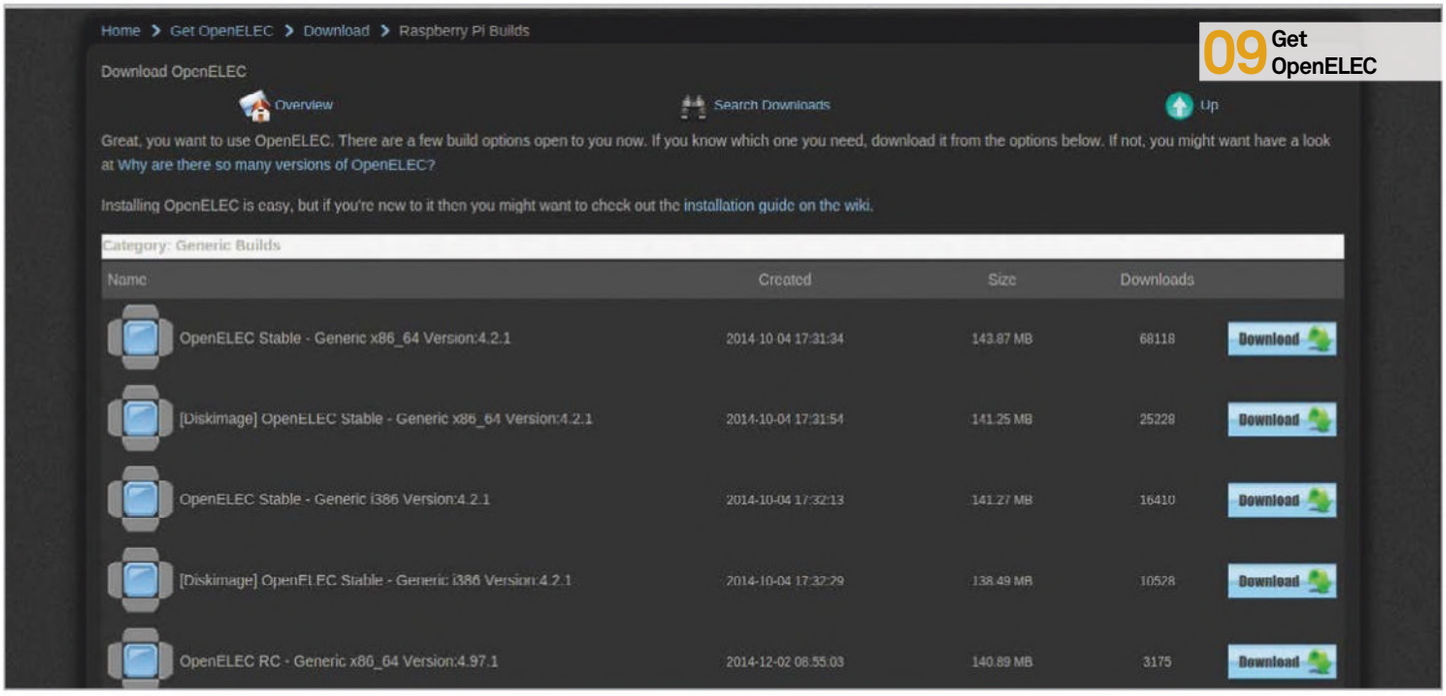
Resources

OpenELEC http://openelec.tv

Some of the best commercial home theatre PCs and media centres run on pre-established and/or open source software. People like to have a familiar interface and to do as little as possible to get their content working. This is why something like Kodi – formerly XBMC – is so popular, as due to years of development, use, testing and maturing it's extremely easy to use and will do/play many things without any extra setup.

A lot of these solutions can lack customisation though, and aren't as open as what you can create yourself in Linux. What we'll show you in this tutorial is all the tools you'll need to create your own dedicated Linux media centre that can power your TV and watch all your shows, or even play your music library in a pinch.

We'll be using Kodi to do this, but a lot of the tips can apply to any other HTPC software you would like to use.



09 Get OpenELEC

01 Choosing your hardware

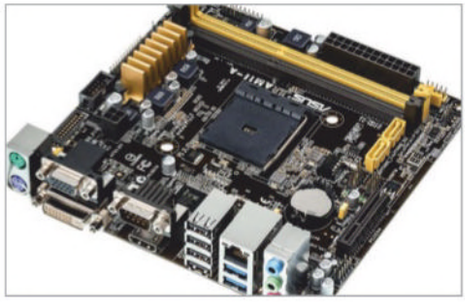
What's going to live under your TV – a custom-built machine or a more expensive mini PC? A mini PC will more than likely be smaller, but you can still make pretty small systems. You can also use a Raspberry Pi, but we're going to concentrate on more traditional x86 solutions.

02 Mini PC choices

There are plenty of mini PCs you could choose for the task – in the past we've used CompuLab's range of IntensePCs and MintBox's for the job, but Zotac and Gigabyte offerings are also up to the task and include a Blu-ray or other disc drive capable of playing your physical library.

03 Getting a case

If you plan to build a HTPC yourself, getting a small case is near essential. While re-using old parts and cases is a good way to do this quickly and cheaply, the resulting product will require a lot of room. Check out slim, mini-ITX chassis for an easy way to get a slimline system that you can readily find components for.



04 Motherboard options

Mini ITX boards are relatively cheap and you can get decent enough ones pretty easily from your regular component supplier. We recommend looking for AMD chipsets that include onboard graphics – these are perfect for 1080p video, thanks to modern codecs and hardware decoding.

05 Other components

You'll also need RAM and a CPU – try and aim for at least 2 GB of RAM, however you should prioritise the RAM over the CPU as the graphics will be doing most of the heavy lifting in the system.

06 Power Supply

Look for green power supplies but don't skimp on the wattage if you can help it. A system like this should have a low idle draw but while decoding high-quality content it may need a lot more than usual. This also allows you to easily upgrade in the future.

07 Repurposing old parts

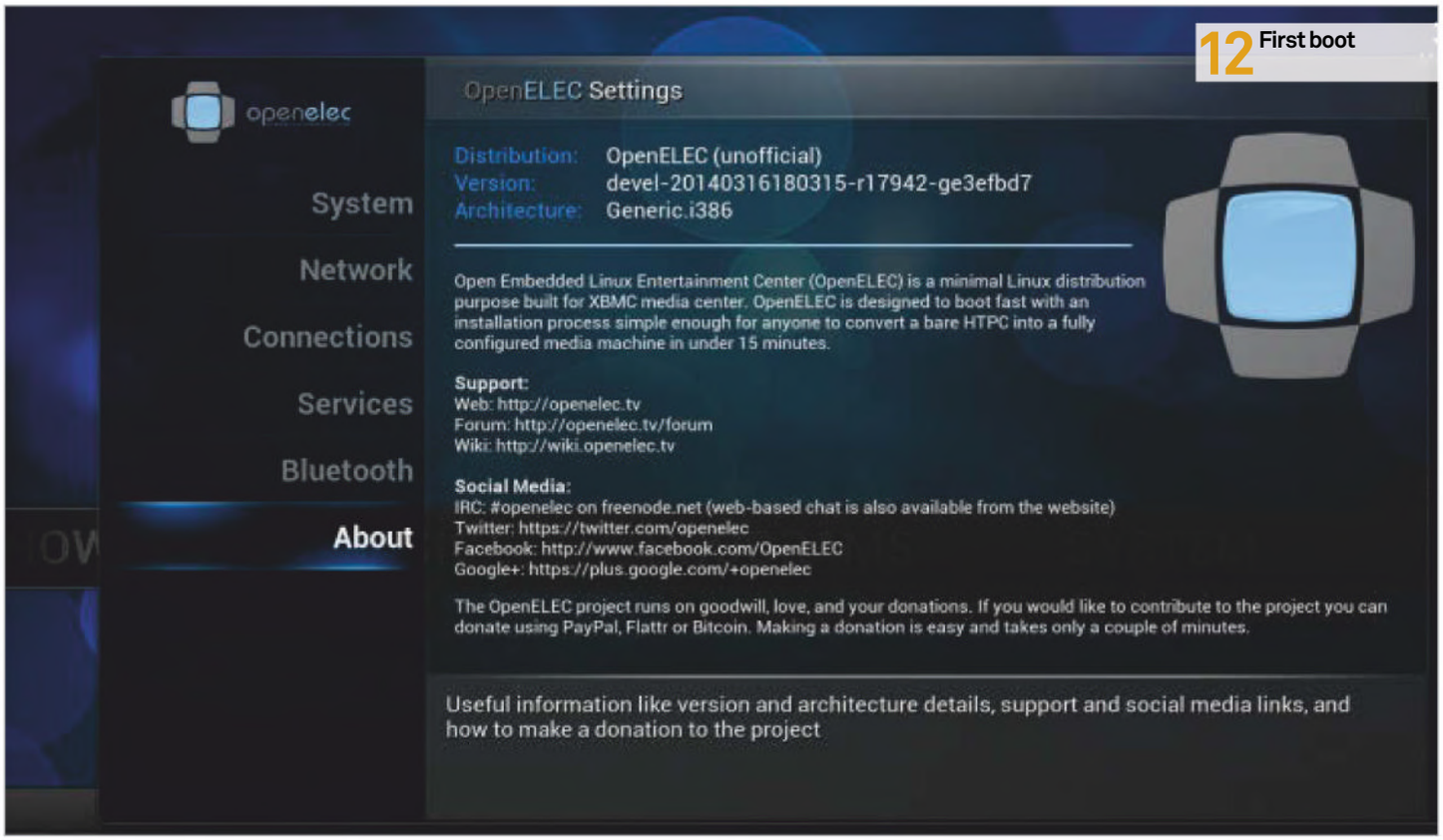
If you have old components lying around, you may well be able to put together a similar system without the need for going with brand new kit. The minimum requirements are tied to the graphics more than anything else, so as long as you have an Nvidia 8500 GT and better, or a Radeon HD 5400 series GPU or newer you should be able to run Kodi fine.

08 Remote control

Most IR receivers and universal remotes will work with the Kodi software, thanks to pre-built modules in the Linux kernel and its software. We'll cover alternate control methods later on.

09 Get OpenELEC

The easiest and probably the best route for setting up Kodi is to grab OpenELEC – it's created by some of the Kodi dev team and is stripped down to the bare essentials to make sure that OpenELEC works on its target hardware. Grab your relevant ISO from http://openelec.tv.



10 Live stick

Unzip the files and open up the terminal. Use **cd** to navigate to the OpenELEC folder, and then insert the stick you'll use to create the live installer. Find out its designation with **sudo fdisk -l** and then set it up with:

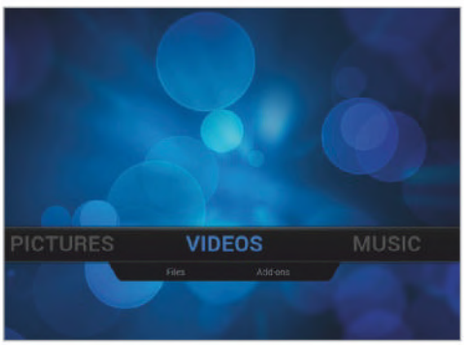
```
sudo ./create_installstick /dev/sdX
```

11 Installation

Once the live media is created, you'll need to insert it into your intended HTPC and switch it on, looking for the option to boot from the stick itself. You'll be asked how you want to install OpenELEC – use the Quick Install, assuming this is a dedicated, completely untouched system, and follow the prompts to install and reboot.

12 First boot

OpenELEC will boot into Kodi/XBMC and now it's time to do some configuring. First, you should look at the internet options in the OpenELEC Configuration Utility located in the Add ons section. This will let you set up wireless internet as well as a few other things.



13 Add networked media

Go to the Videos/Music tab and find the Files option – here you can add media from networked sources via Samba, UPnP or with a direct address to something like an NFS partition. You can then choose a scraper which will add art and names to any files you may have available.

14 Enjoy your shows!

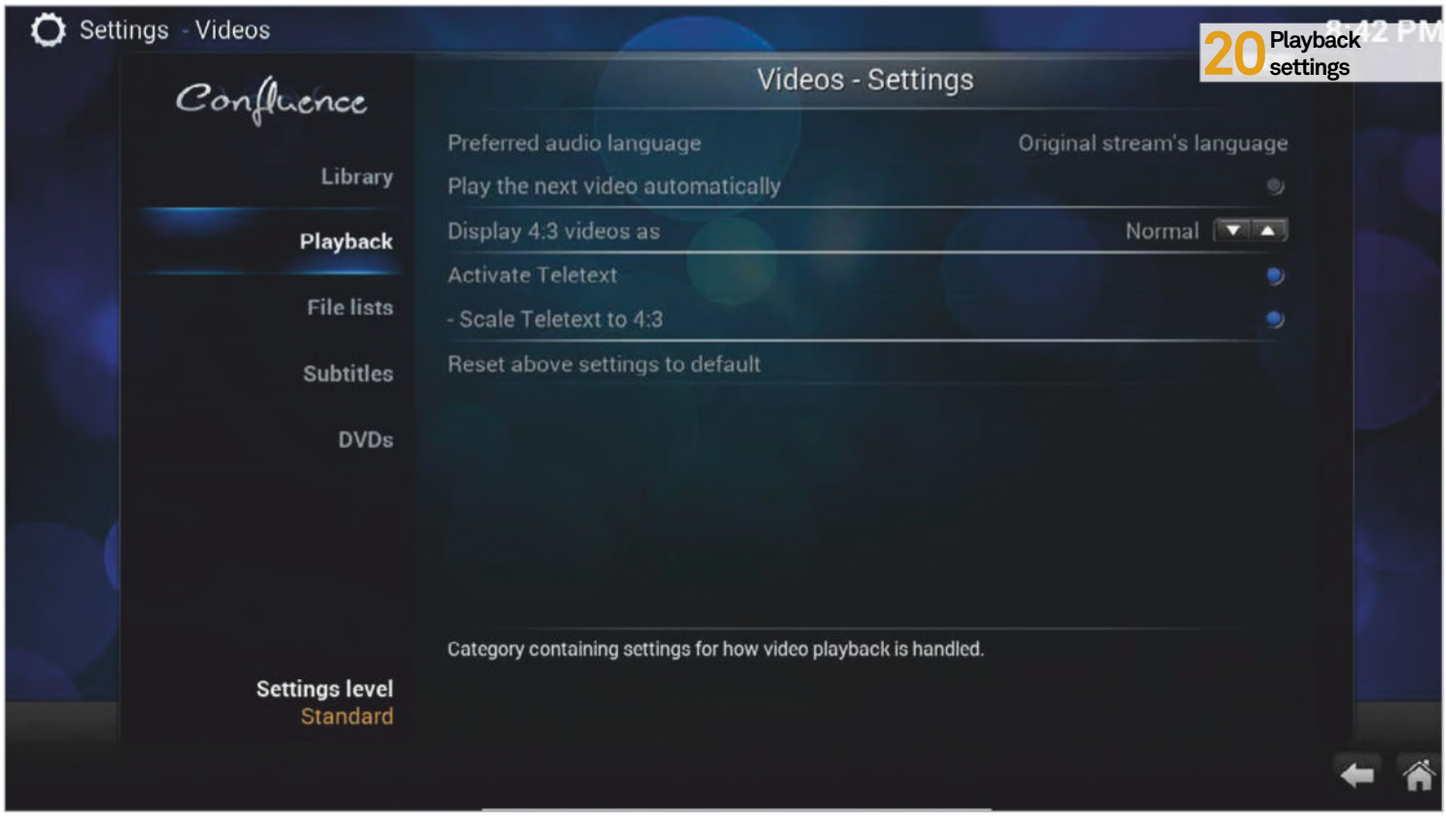
Once your network is set up, and your media is added from around the network, you can start watching or listening to anything you want. It doesn't take much time at all to set up a proper media centre PC, as we've shown you here. Plus, you don't even need to do much more in the way of maintenance to keep using it like this. Kodi has a lot of great extra features though, so continue on for ways to get the most out of your HTPC.

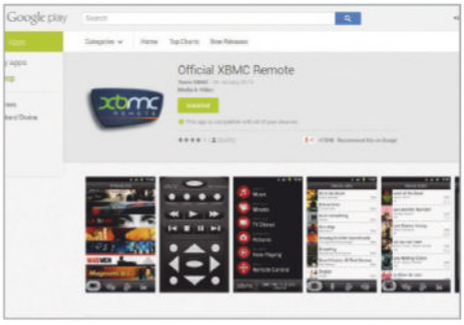
15 Internal storage

Keeping files on internal storage is a good way to give yourself have constant access to them – especially shows that you will watch regularly or people in your household (like kids with cartoons, for example) will watch over and over. These will be automatically added to your list without you needing to point Kodi at their location.

16 USB storage

Accessing the internal storage may be a little difficult for some, but a USB stick or portable hard drive will easily connect to the system and will be instantly added to your video library like anything from the internal storage as well.



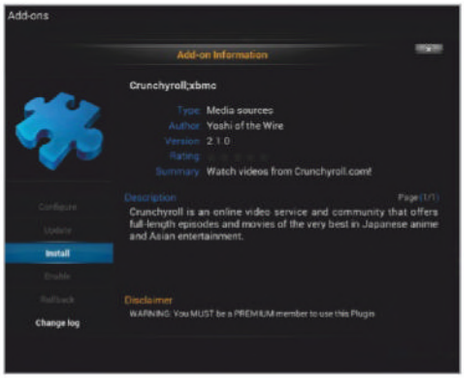


17 Smartphone remote

As well as the physical remote that you (may have) set up, you can also use an Android smartphone as a wirelessly connected remote, using the HTML options you can also access in a browser. The (XBMC-branded) app can be downloaded from the Play Store here: **https://play.google.com/store/apps/details?id=org.xbmc.android.remote**

18 Other control options

As OpenELEC is built on Linux, it comes complete with various drivers that allow you to use various game controllers – PS3 and wired 360 pads in particular. This can help you in a pinch and may be best if you plan to dual boot your HTPC.



19 Add-ons

There are a lot of video, music and program add-ons for Kodi left over from XBMC. Find the Add On option under each category to find a list of plug-ins that can be instantly added to your HTPC.

20 Playback settings

In the main settings you have plenty of options to tweak playback to give you a better experience. Do you want to specify an audio or subtitle language for any videos? What about aspect ratio for 4:3 shows? This can all be changed and configured in the settings, making it easy for you to tailor your experience to what you want.

21 Live TV

You can watch and record Live TV through Kodi as well, although you'll need to get an extra component to attach aerial input. There are EPG features you can activate, along with behaviour for the PVR functionality that you can also change.

22 Keep watching

There's a lot to discover with Kodi, but this should get you started, leaving you plenty of room to expand in the future. As for upgrades, the hardware should last longer than most systems, and OpenELEC has an in-built software updater as well.

Manipulate and convert data in R

Advisor

Mihalis Tsoukalos is a Unix administrator, a programmer (for Unix and iOS), a DBA and also a mathematician. He has been using Linux since 1993

Resources

R project r-project.org

RStudio bit.ly/1VLmlgA

Zoo package bit.ly/1Ov0whd

Chron package bit.ly/1LhClzq

Learn how to use R to easily and effectively manipulate various kinds of data

The R software deals, processes and visualises data all the time – it is something that can give value to your data. However, what is the point of visualising data if you cannot transform, manipulate and change your data whenever you want, any way you want?

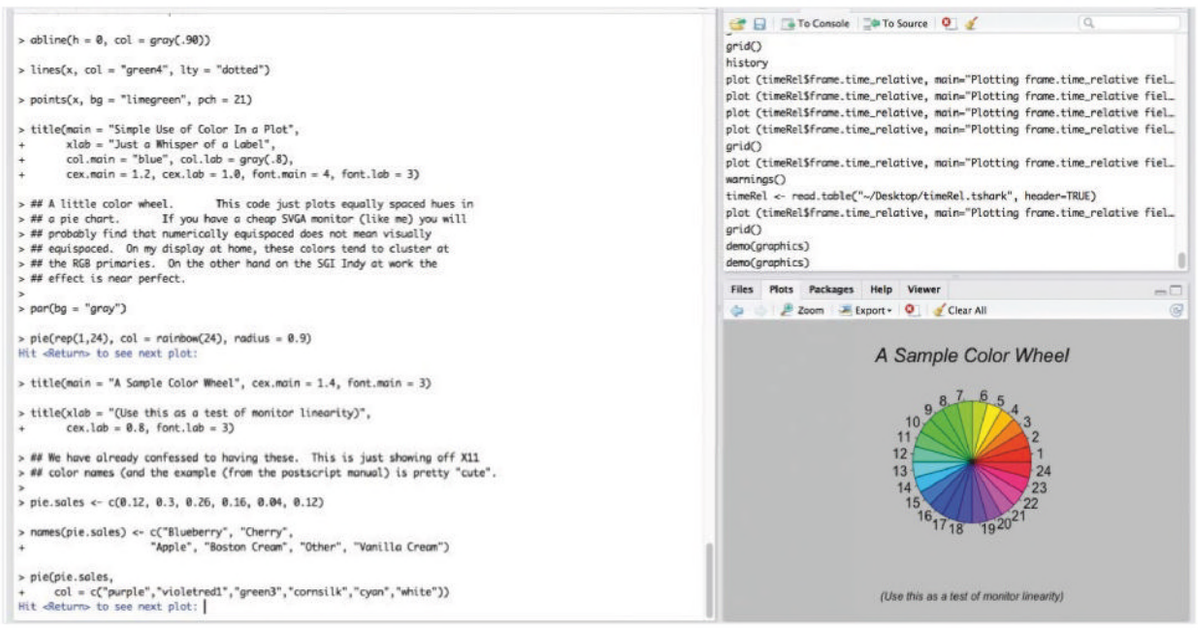
This article will teach you many different ways to manipulate and transform your stored data within R. This is very important because most of the time the information you want is hidden somewhere and waiting for you to find it. Additionally, when you have problems with data, it is often a matter of data being in the wrong mode or class in relation to the task you are trying to perform.

The first thing that you should remember is that the index of the first element in R data types is one instead of zero.

If you would prefer to use an IDE instead of the command line version of R then you can download the open source version of RStudio from bit.ly/1VmlgA, but RStudio is not necessarily required in order for you to follow this tutorial. Nevertheless, you will still benefit from knowing how to install, run and quit R, as well as how to type commands to its command line environment.

01 The various types of data in R

R supports various types of data, and each type has unique c properties. This part of the tutorial will introduce you to the most important data types for handling groups of data. A list is a generic vector containing other objects and a vector is a sequence of data elements of the same basic type. A data frame is a list of vectors of equal length that is primarily used for storing data tables. It is used a lot in R and is equivalent to the concept of a table. An array is a multidimensional object. A matrix is a two-dimensional array that contains numeric data; it has rows and columns.



This is the environment of RStudio, a powerful and useful IDE for R

The **mode()** function returns the mode of an object whereas the **class()** function returns the class of an object. Both functions can be very useful when you do not know the kind of R object you have.

02 Import data in R

Usually, you do not write your data yourself; most of the time you will get your data from external sources, including text files, databases and the Internet. The reading of external data is made with the help of the **read.table()** function that returns a data frame. If you want to read an external text file, you can use the following command:

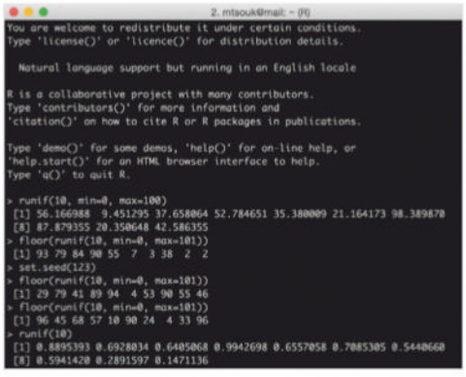
```
> myData <- read.table("./data.txt", header=FALSE)
```

If you set the header parameter to **TRUE**, it is assumed that the first line of your input will contain the names of the variable.

Remember that depending on the format of the external data, you should use the appropriate data type to hold your data or transform it to the desired format.

03 Create and use random data

Being able to get and work with random numbers is great for testing various methods and techniques. Therefore this step will teach you ways to get random data in R. The following returns ten floating-point numbers from 0 to 100:




```
> runif(10, min=0, max=100)
```

The next variation returns ten integers between 0 and 100:

```
> floor(runif(10, min=0, max=101))
```

Please note that the maximum number cannot be returned, so putting 101 as the value of the **max** parameter ensures that the integer number 100 can be returned.

In case you want to generate the same sequence of random numbers again, you can use the **set.seed()** function as follows:

```
> set.seed(123)
> runif(10, min=0, max=100)
```

As you can see in the image, you must run **set.seed()** before executing the **runif()** function.

There is a function called **sample** that shuffles the contents of an existing vector into a random sequence without changing the actual numeric values:

```
> a <- floor(runif(3, min=0, max=100))
> sample(a)
[1] 57 89 10
> sample(a)
[1] 89 10 57
```

If you put a second argument to **sample**, you can specify the size of the sample that is going to be returned.

04 Deal with text and match

Although most of the article will be about manipulating numeric data and dates, this step will talk about text manipulation in R and how to do things with text.

The **as.numeric()** function converts a character string to its numeric value. If there exist invalid characters, you will get an error message. You can concatenate two strings as follows:

```
> paste("Linux User", "Developer", sep=" and ")
[1] "Linux User and Developer"
```

Similarly, you can link the strings from two columns of a matrix into a new column. If the matrix has

```
> myData <- read.table("./data.txt", header=FALSE)
> system("cat ./data.txt")
10   a1   c1
20   a2   c2
30   a3   c3
40   a4   c4
50   a5   c5
> myData
  V1 V2 V3
1 10 a1 c1
2 20 a2 c2
3 30 a3 c3
4 40 a4 c4
5 50 a5 c5
> class(myData)
[1] "data.frame"
> mode(myData)
[1] "list"
>
```

Here's the output from Step 2, where we read our table for data

Extra R packages

Although R is a very capable package with a great programming language, it's true power comes from the numerous R packages. Whenever you have a problem you want to solve, check if there exists a package that can help you do your job. You can start from **cran.r-project.org**.

three columns, then a new fourth column will be created in the returned matrix; it is your job to store the new matrix.

R also supports pattern matching for text variables. So, should you wish to get all columns that contain the text "Error", you should run the following command that utilises the **grep()** function:

```
> errorMessages <- c("I/O Error", "Network Error",
  "Data not found", "TCP/IP not working", "Data Error")
> errorMessages[grep("Error", errorMessages)]
[1] "I/O Error"      "Network Error" "Data Error"
```

Another truly important function is **match()**, which answers the question "where can the values of the second vector be found in the first vector?" The **match()** function can be better understood with an example, like so:

```
> firstV <- c(1, 2, 4, 5, 9, -1, 0, 4, 5)
> secondV <- c(0, 1, 2, 0, -1, 9)
> match(firstV, secondV)
[1]  2  3 NA NA  6  5  1 NA NA
```

Match() returns a vector of subscripts that belong to the second vector and has as many elements as the first vector. If an element of the first vector cannot be found in the second vector, then NA appears in its place. Otherwise, the index of the first occurrence of an element is returned instead of NA.

05 Deal with dates

Dates and times are special kinds of data and therefore need special treatment when dealing with them. It is advisable not to store dates and times as plain text because it is difficult data to manipulate. However, occasionally you will need to convert a date or time stored as a string into a more appropriate format.

R offers the built-in Date, POSIXlt and POSIXct classes for storing dates and times, as well as the zoo and chron packages. This article will only deal with the built-in types.

The general principle is that if you are using a nonstandard format, you will have to specify it. Moreover, R enables you to do calculations with dates,

POSIXct (ct: Calendar Time) is the best class when you have times in your data; this class also lets you specify the timezone of a date. The POSIXlt (lt: Local Time) class enables you to easily extract specific components of a time. It is important to remember that POSIXlt objects are lists. If you only have dates then use the Date class. Type **help(DateTimeClasses)** to get more information about date and time classes.

06 More about states and times

The **strptime()** function enables you to convert a factor or a string into a date – the user must provide a format statement in double quotes to inform R about the structure of the input. A factor is of mode numeric and class factor.

The **difftime()** function can also help you find the difference between two dates.

It is very important to use the correct code when parsing dates and times in R: **%Y** is for four-digit years whereas **%y** is for two-digit years. Use **%d** to declare the day of the month and **%m** for declaring the month as a decimal number. Use **%B** as the code for the full name of a month and **%b** for the abbreviated name of a month.

It is necessary to try things using small samples before working with real data, especially with dates and times where it is easy to make small typos that can create big errors.

07 Data tasks

This part will show you how to do some simple things with your data. Suppose that you have a data frame named **aDataFrame**. If you want to get a single element from the data frame you should run the following command:

```
> aDataFrame[4,3]
```

The previous command will get you the element from row four and column three. Similarly, you can get its first column as follows:

```
> aDataFrame[,1]
```

Should you wish to get its second row, you should execute the following command:

```
> aDataFrame[2,]
```

To get the first three rows of a data frame, you can use the next notation below. This can also be applied for getting the first three columns:

```
> aDataFrame[1:3,]
```

08 Advanced data tasks

The following command will define a data frame with three columns:

```
> myDataFrame = data.frame(v1=c(1,2,3,4,5),
  v2=c(0,1,2,3,4), v3=c(-1,-2,-3,-4,-5))
```

The following command will add the same number to all data frame elements:

```
> myDataFrame + 5
```

As the columns in myDataFrame have names, the following two commands will both subtract the number five from the first column of the data frame:

```
> myDataFrame[, 1] - 5
[1] -4 -3 -2 -1  0
> myDataFrame$v1 - 5
[1] -4 -3 -2 -1  0
```

The following command adds ten to all elements of the first row:

```
> myDataFrame[1, ] + 10
```

The following command adds minus one to the first column, zero to the second and one to the third column of myDataFrame:

```
> myDataFrame + -1:1
```

Remember that as long as you do not assign the result of an operation to the **myDataFrame** variable, the original contents of myDataFrame will not change.

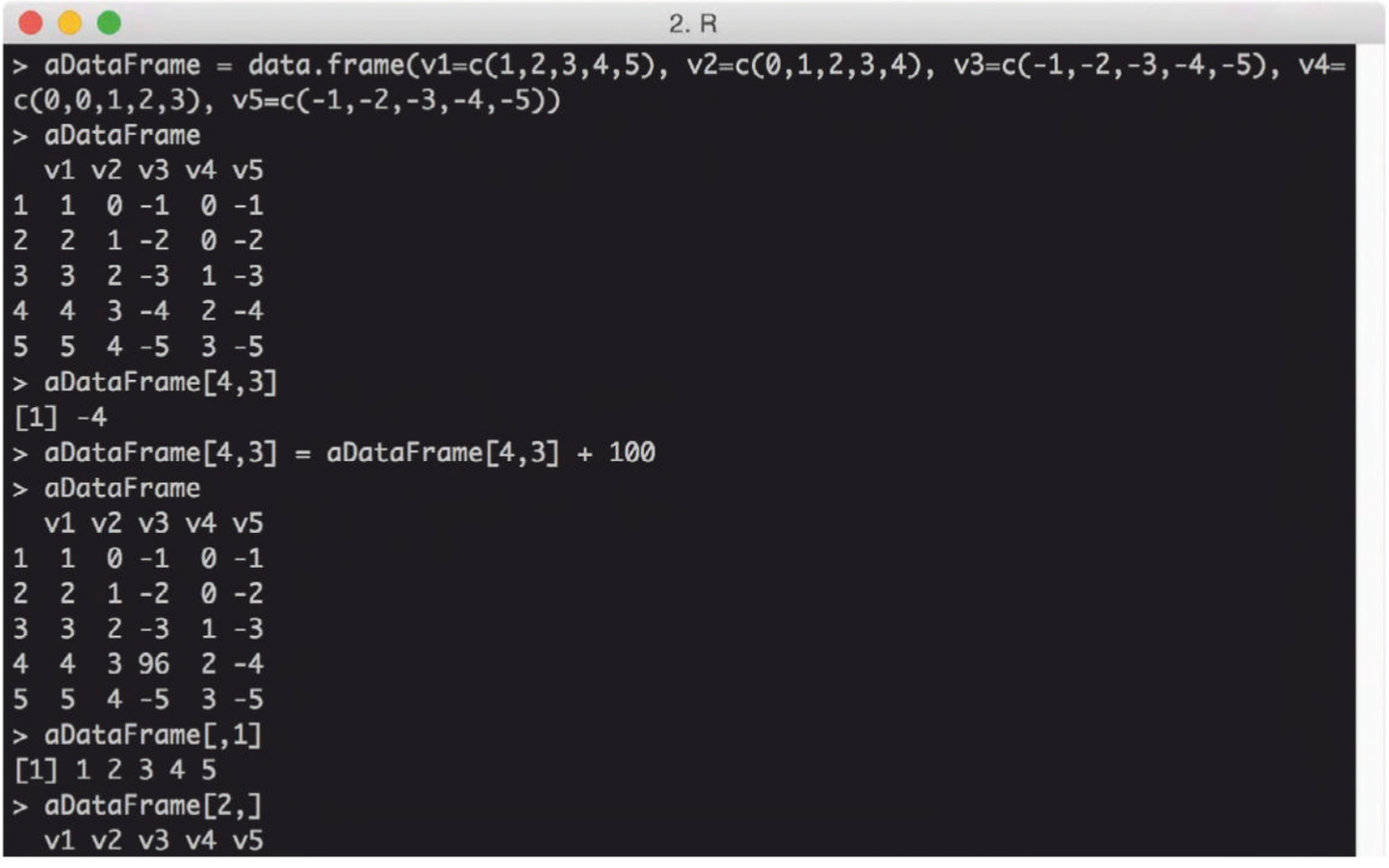
The next command creates a new column named sum to myDataFrame that contains the sum of all values of each row:

```
> myDataFrame$sum = myDataFrame$v1 +
  myDataFrame$v2 + myDataFrame$v3
```

As you can understand, this operation does alter myDataFrame by adding a new column to it. Using the same method, you can perform any kind of calculations you want.

09 Convert between various data types

This step will show you how to convert between lists, data frames, vectors and also matrices. Given a matrix named **myMatrix**, you can convert it into a list using the following simple command:



R enables you to apply operations to specific rows and columns of a table

```
> myList <- as.list(data.frame(t(myMatrix)))
```

Given a list name **myList**, you can convert it into a matrix with this command:

```
> anotherMatrix <- matrix(unlist(myList),
ncol = 2,
  byrow = TRUE)
```

The previous command requires that you manually specify the number of columns of the new matrix because you are converting a linear type into a two-dimensional one. You can transform a vector named **aVector** into a list as follows:

```
> anotherList <- as.list(aVector)
```

You can convert a list named **anotherList** into a vector with the help of the **unlist()** function:

```
> aNewVector = unlist(anotherList)
```

10 Save and load your work

So far, we have shown you how to change your data in various ways. The final step will describe how you can save your data and load it the next time you decide to run R. You have the option to save your current R session, including all defined variables, as follows:

```
> save(list = ls(all=TRUE), file =
"R25Aug2015")
```

The operating system will create a file named R25Aug2015 that you can load afterwards as follows:

```
> load((file = "R25Aug2015"))
```

The following command can be used for removing all currently defined types of R objects:

```
> rm(list=ls())
```

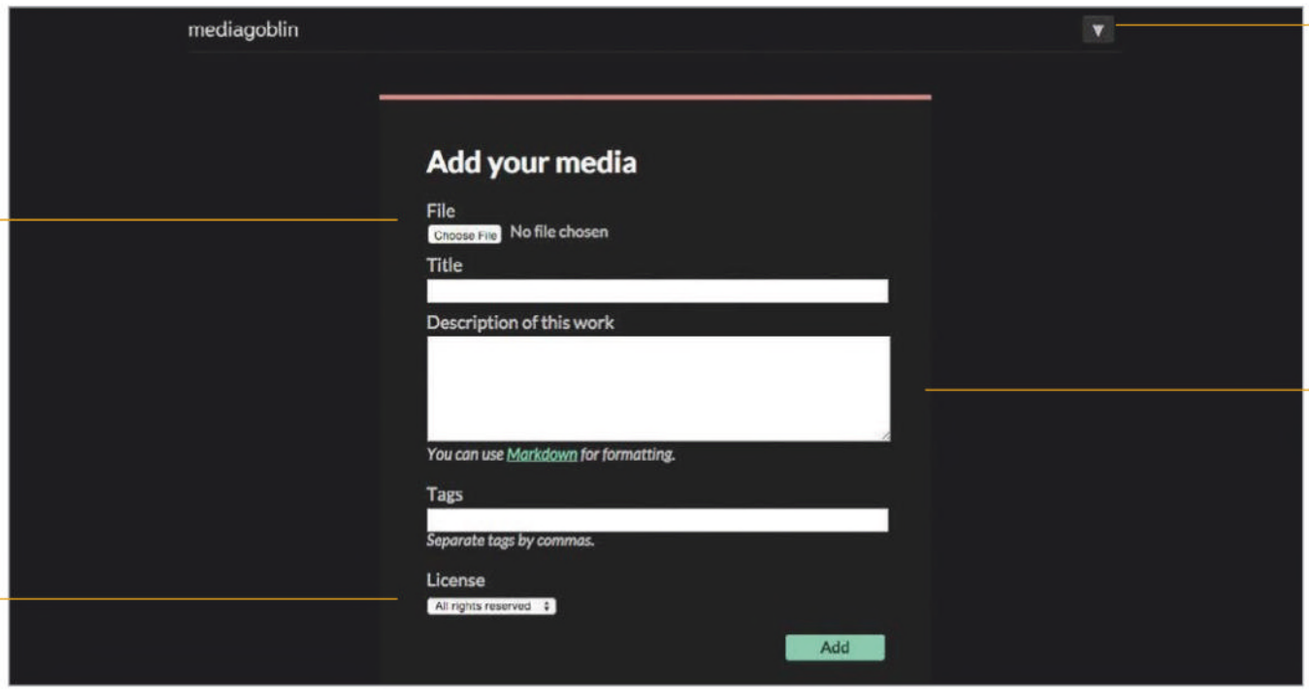
R is an endless subject and you can only learn more about it by practising and exploring your options, so start experimenting today and give your data many new meanings!

Get help

R offers a plethora of ways to get help about functions. If you type a question mark at the R prompt, followed by the name of a function, R will return something similar to a UNIX man page:
> ?read.table
The following command will return working examples for the read.table function:
> example(read.table)
The next command will show you an impressive demo of the graphical capabilities of R:
> demo(graphics)

Here, you can add new files or create a collection of media files. Files can be assigned to collections at any point of time

Update account settings here and track media files using the Media processing panel, which shows details about uploads



All creative commons licences are available here to assign under an available media file

This section lets you upload a media file and update the relevant information regarding that file

Host your own media gallery with MediaGoblin

MediaGoblin provides a way to share videos, photos or audio recordings with your loved ones, without worrying about privacy

Advisor

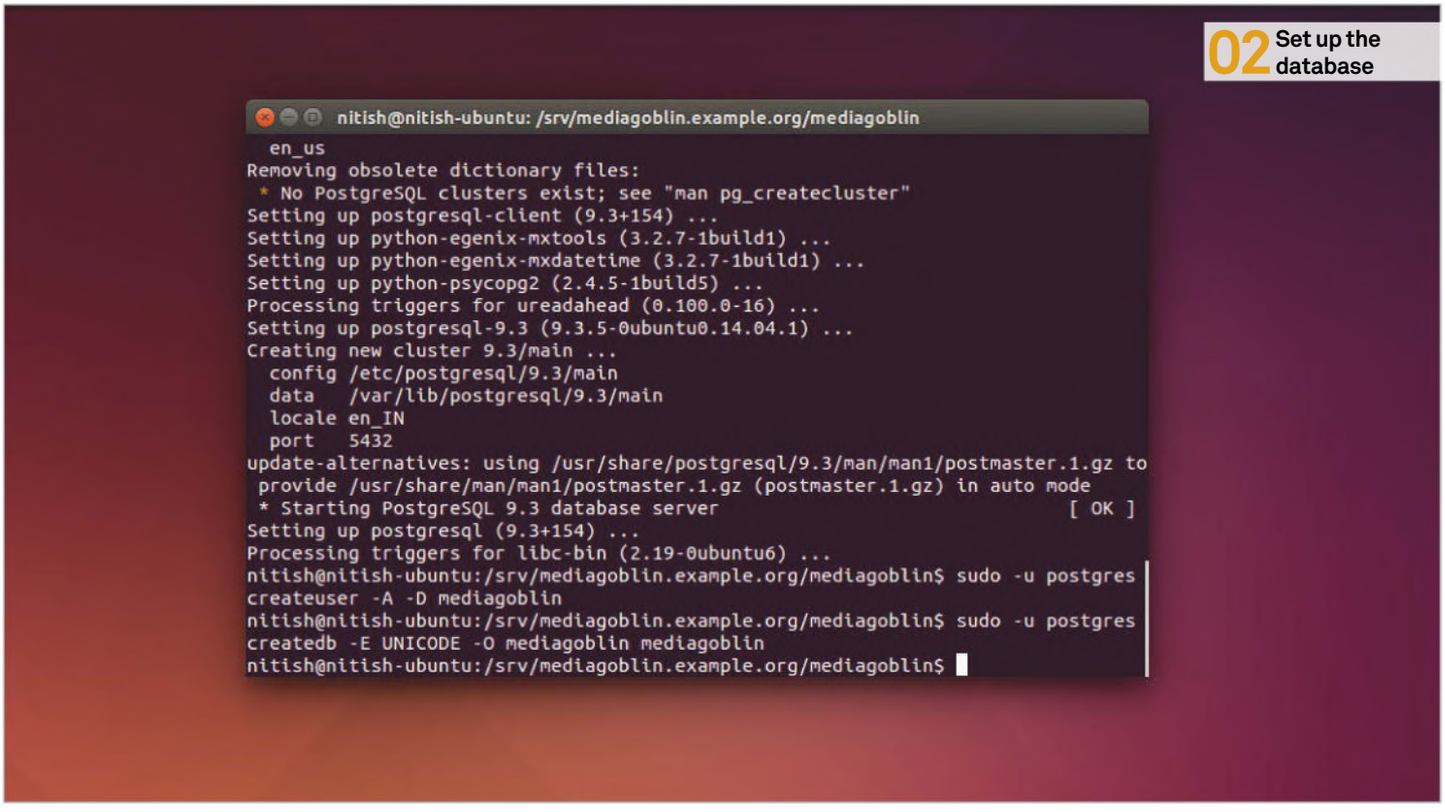
Nitish Tiwari is a software developer by profession and an open source enthusiast by heart. As well as writing for leading open source magazines, he helps firms set up and use open source software for their business needs

Resources

MediaGoblin home page
mediagoblin.org

YouTube is not only a website anymore, it's become a phenomenon. Millions of hours are spent – or wasted – daily in watching videos of cats, dogs and humans doing strange things. With the predictive playlist appearing just after you finish a video, it is sometimes very difficult to close the window and you go on and on, watching one video after the other. But YouTube is a dangerous place for personal videos and other media that you don't want strangers to access. Though it has an option to make your videos private, you don't really know how private it is. So we need to find a solution that has the perfect match of convenience and privacy.

This is where MediaGoblin comes in. This open source tool can help you organise, host and stream media from your own PC without having to worry about privacy. If you are a power user, you can also have it run on a web server and let other people add their media. There are a range of other useful features available too, like tags and collections to name just two. In this tutorial, we will begin with the steps to install MediaGoblin on Ubuntu and then proceed to get it working and see it in action. We have used Ubuntu 14.04 as the host system and MediaGoblin's clone from their Gitorious repo.



01 Sort out dependencies

MediaGoblin is a full-fledged media-streaming platform and therefore it has a few software dependencies that you will need to take care of before installing MediaGoblin. Let's take a look at these dependencies and how to install them.

- **Python 2.6 or 2.7** This interprets the MediaGoblin source code.
- **Python-lxml** Binds certain C libraries to Python.
- **Git** For downloading and updating the repository.
- **SQLite/PostgreSQL** This is where everything is stored. SQLite is the default option and works fine for small set-ups, but you need to use PostgreSQL if you expect more users.
- **Python Imaging Library** This adds image-processing capabilities to Python interpreter.
- **virtualenv** This is used to create isolated Python environments.

You can install all these on a Debian based system, using the apt-get command. It can be done with a single command:

```
sudo apt-get install git-core python
python-dev python-lxml python-imaging \
python-virtualenv
```

02 Set up the database

As we said before, the default SQLite database doesn't perform well for deployments involving more than two or three users. So, if you are planning to have more than three users, it's recommended to use the PostgreSQL database. To set it up for MediaGoblin, first download and install the packages using apt-get:

```
sudo apt-get install postgresql
postgresql-client python-psycopg2
```

Note that it has other required packages too. The installation process creates a user with sufficient privileges to handle the database, but keeping security in mind we will create a separate user for MediaGoblin. To create the new user, type:

```
sudo -u postgres createuser -A -D
mediagoblin
```

Once the user is created, create the database:

```
sudo -u postgres createdb -E UNICODE -O
mediagoblin mediagoblin
```

Here the first 'mediagoblin' is the user name and the second is the name of database.

03 MediaGoblin user

You'll have noticed that we didn't add a password for the user named mediagoblin. So how does the system authenticate the user? This is done via the local Unix authentication. Local Unix authentication allows a system user to connect to any PostgreSQL database on the system without a password. To enable this, you need to create a system user with same name as the PostgreSQL database user. So we now need to create an unprivileged system user named mediagoblin. Note that the user can be underprivileged because MediaGoblin doesn't need any privileges to run. This also makes the system more secure. Run this:

```
adduser --system mediagoblin
```

You can't login to this account but a switch using:

```
sudo -u mediagoblin /bin/bash
```

You can then use this user account for all further steps.

"Note that it has other required packages too"

04 Install MediaGoblin

You need to create a working directory for MediaGoblin. This is where the git repository will be downloaded. Create the directory using:

```
sudo mkdir -p /srv/mediagoblin.example.org && sudo chown -hR mediagoblin /srv/mediagoblin.example.org
```

As you can see, we create the directory with elevated privileges (root) and then change the owner to our underprivileged mediagoblin user. Let's now clone the MediaGoblin repo to this folder. First switch to the mediagoblin user and then change the directory to the working directory we just created:

```
cd /srv/mediagoblin.example.org
```

Now start cloning:

```
git clone git://gitorious.org/mediagoblin/mediagoblin.git
```

Move to the 'mediagoblin' folder – `cd mediagoblin`. Initialise the repo and then fetch the data:

```
git submodule init && git submodule update
```

You'll notice that we didn't take code from the stable revision but instead the master branch of the git repository. MediaGoblin is under rapid development so it makes sense to use the master, at least until a consistent release.

05 Install Virtualenv & others

MediaGoblin uses virtualenv – a tool to help manage the dependencies by creating isolated Python environments. It's already available in the package, so set it up by using:

```
(virtualenv --python=python2 --system-site-packages . || virtualenv --python=python2 .) && ./bin/python setup.py develop
```

If you are feeling adventurous, you can also try the experimental deploy system (shell script) instead of the earlier command:

```
./experimental-bootstrap.sh && ./configure && make
```

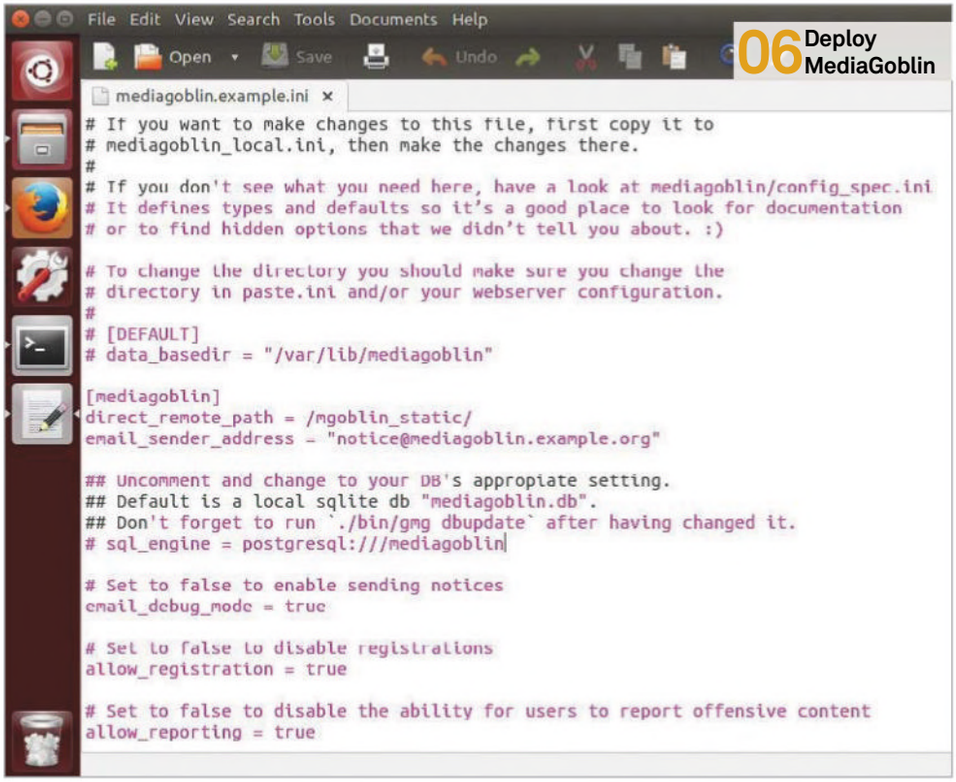
This script sets up virtualenv and also helps you keep it updated by running **make update**, but as per the developers of MediaGoblin, this is still under development and may break. To update the codebase at a later point of time simply run:

```
git submodule update && ./bin/python setup.py develop --upgrade && ./bin/gmg dbupdate
```

You also need to install Flup before the setup concludes. Install it using:

```
./bin/easy_install https://pypi.python.org/pypi/flup/1.0.3.dev-20110405
```

We will see more on Flup and FastCGI later.



06 Deploy MediaGoblin

Now that dependencies are set up and MediaGoblin is installed, we will edit the MediaGoblin configuration files – specifically the mediagoblin.ini file located inside /srv/mediagoblin.example.org/mediagoblin. Here are the changes required:

- Set `email_sender_address` as the ID you want to use for sending system mails.
- Uncomment the line `sql_engine = postgresql:///mediagoblin` if you are using PostgreSQL.
- Edit `direct_remote_path`, `base_dir` and `base_url` as per the root of virtual host.

Now update the database using `./bin/gmg dbupdate`. This populates the database with MediaGoblin data structures. Finally, test the MediaGoblin server using:

```
./lazyserver.sh --server-name=broadcast
```

You should now be able to connect on your browser port 6543.

07 Flup and FastCGI

MediaGoblin uses FastCGI for deployment and FastCGI needs a server. So we need Flup. We already installed Flup in step five. Later you will learn a FastCGI setup with an Nginx server to serve MediaGoblin pages.

"MediaGoblin is under rapid development so it makes sense to use the master"

FastCGI is a protocol to interface interactive programs with a web server – it's an improvement over CGI (common gateway interface). CGI, while easy to implement, had problems in scaling since separate processes were created for each web req uest – a huge overhead for the host OS. FastCGI solves this by using persistent processes to serve series of web requests; moreover, these processes are owned by FastCGI server (Flup in our case) and not the web server. This de-couples webserver and FastCGI server, allowing effective scaling. Now any server that supports FastCGI can be used for MediaGoblin. Nginx is a good option because of easy configuration and setup.

08 Nginx setup

Nginx has been slowly rising in the ranks of the web server of choice and is currently one of the most used web servers. An acronym for Engine X, it is a high-performance HTTP server. It does support a lot of other protocols too but those are out of scope for us here. Let's go straight to the server set up. Create a configuration file at /srv/mediagoblin.example.org/nginx.conf and create a symbolic link into a directory that will be included in your nginx configuration with one of the following commands (as the root user):

```
ln -s /srv/mediagoblin.example.org/nginx.conf /etc/nginx/sites-enabled/
```

This way, a change in one file automatically reflects in the other. You need to then add the contents to the configuration file, as shown in the screenshot below. Remember to change the fields as per your local paths. Once done, restart nginx using `sudo /etc/rc.d/nginx restart`. If everything goes well, start MediaGoblin using:

```
cd /srv/mediagoblin.example.org/mediagoblin/ ./lazyserver.sh --server-name=fcgi fcgi_host=127.0.0.1 fcgi_port=26543
```

Visit **mediagoblin.com** to see an example MediaGoblin gallery in action.

09 MediaGoblin home

The setup process is a little lengthy, and for the novice user it may seem a complex task, but the steps are simple and you just need to follow them one at a time. Once you have successfully completed the process, you can enjoy uninterrupted media streaming for you and your loved ones.

The first step after you're ready with your own MediaGoblin instance is to create an account.

08 Nginx setup

```
server {
 ################################################
 # Stock useful config options, but ignore them :)
 ################################################
 include /etc/nginx/mime.types;

 autoindex off;
 default_type  application/octet-stream;
 sendfile on;

 # Gzip
 gzip on;
 gzip_min_length 1024;
 gzip_buffers 4 32k;
 gzip_types text/plain text/html application/x-javascript text/javascript text/xml text/cs

 ####################################
 # Mounting MediaGoblin stuff
 # This is the section you should read
 ####################################

 # Change this to update the upload size limit for your users
 client_max_body_size 8m;

 # prevent attacks (someone uploading a .txt file that the browser
 # interprets as an HTML file, etc.)
 add_header X-Content-Type-Options nosniff;

 server_name mediagoblin.example.org www.mediagoblin.example.org;
 access_log /var/log/nginx/mediagoblin.example.access.log;
 error_log /var/log/nginx/mediagoblin.example.error.log;

 # MediaGoblin's stock static files: CSS, JS, etc.
 location /mgoblin_static/ {
    alias /srv/mediagoblin.example.org/mediagoblin/mediagoblin/static/;
 }

 # Instance specific media:
 location /mgoblin_media/ {
    alias /srv/mediagoblin.example.org/mediagoblin/user_dev/media/public/;
 }

 # Theme static files (usually symlinked in)
 location /theme_static/ {
    alias /srv/mediagoblin.example.org/mediagoblin/user_dev/theme_static/;
 }

 # Plugin static files (usually symlinked in)
 location /plugin_static/ {
    alias /srv/mediagoblin.example.org/mediagoblin/user_dev/plugin_static/;
 }

 # Mounting MediaGoblin itself via FastCGI.
 location / {
    fastcgi_pass 127.0.0.1:26543;
    include /etc/nginx/fastcgi_params;

    # our understanding vs nginx's handling of script_name vs
    # path_info don't match :)
    fastcgi_param PATH_INFO $fastcgi_script_name;
    fastcgi_param SCRIPT_NAME "";
 }
```

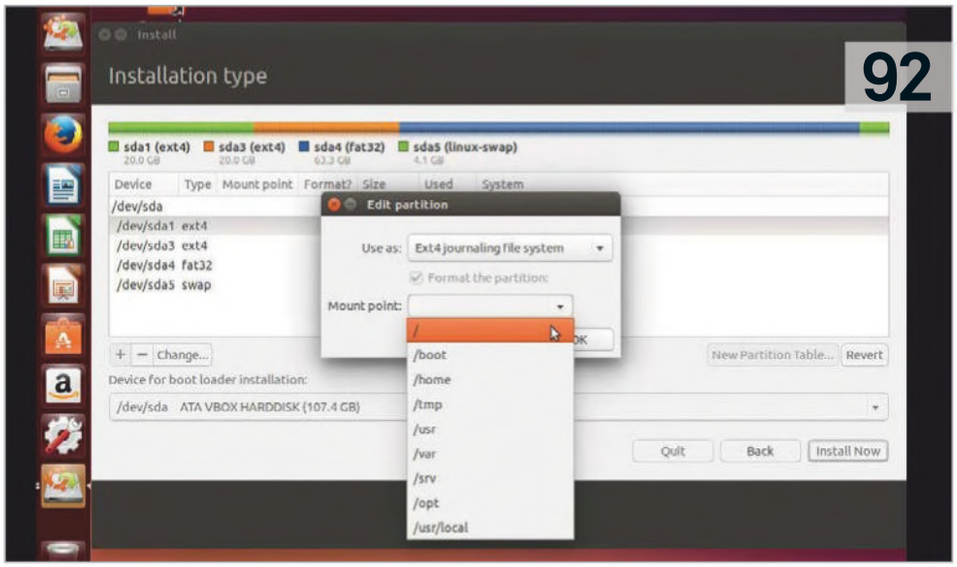
This is because you can browse the collections anonymously but you need an account to upload media. To create an account click on the 'Log in' button on the top-right corner. In the log in page that appears next, click on 'Create one here' to open the user registration page. Fill the details in the registration page and you are good to go! Just log in with the credentials and then click on your user name to be redirected to your profile page. Here you have the options to upload media and manage your profile.

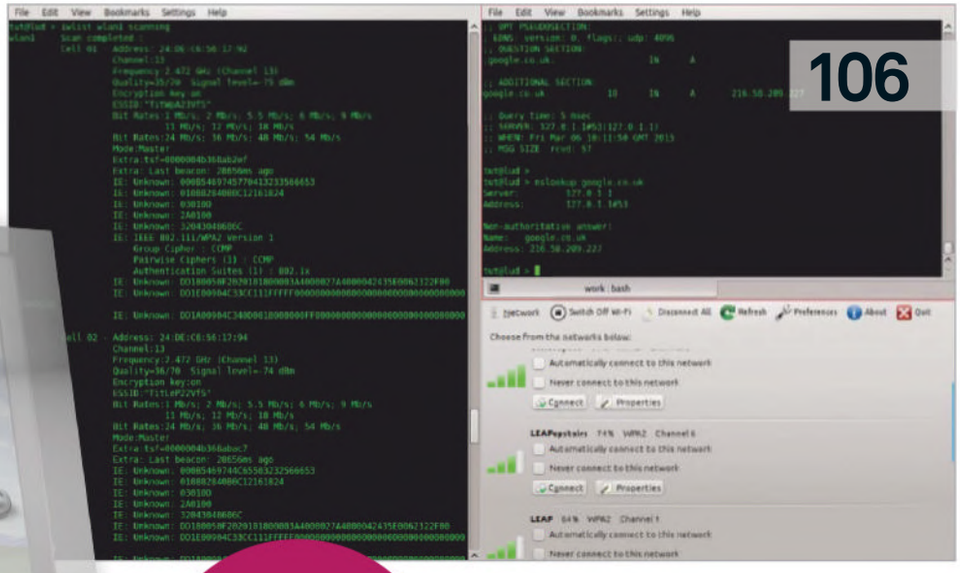
10 Add media

Adding a media file is a breeze, just click on the 'Add media' button on the right side of your profile page. In the next page, upload the file, set the title of the media, add a description, add tags and set the license you'd like to assign to the media. Finally click on the 'Add' button to upload the file. On the left side of the page, you may notice the 'Browse Collections' link. This option lets you browse collections created by other users (if you are on a multi-user environment). A collection is a group of media files logically bundled together, generally to represent an event or other such scenarios. Note that media files can be added to collections at any point in time and not just during the upload. To create a collection yourself, click on the top-right icon to reveal the account related options and then click on 'Create new collection'. You can then add the title and description to add your own collection.

Masterclass

Become a Linux power user





Page 82
50 Critical fixes

"The order has its advantages by more easily tracking what you're installing and where"


Privacy, simplified.
Get AVG PrivacyFix for free
One dashboard for your Facebook®, Link
Blocks over 1200 tr

50 CRITICAL FIXES

LEARN TO FIX **50** OF THE MOST IMPORTANT LINUX PROBLEMS

One of the upsides to using Linux is its vast hardware compatibility that's built into the kernel, with countless modules able to help you boot and run on just about any piece of computing hardware from the last 20 years. It's also fairly rock-solid and continually supported by a massive community to improve it and make it better and better.

Having said that, though, there are always little problems. Some may lie in software bugs or quirks in the kernel, problems with packages and software, and even random defects caused by the hardware itself. Like any complicated piece of equipment, Linux and its distros can go wrong in many ways. Learning how to fix these problems can be confusing for some.

In this feature, we've compiled what we believe to be 50 of the most common problems and solutions to your day-to-day Linux issues – ones that will help you out when either the inevitable happens or you are simply helping someone out with some troubleshooting.

RESOURCES

GParted
The graphical partition editor that we always recommend, it's an easy way to manage, delete and change your current hard drive set up. It comes installed by default on a lot of live CDs and there's also a specific live release of the software that has a few extra hard drive management features available on it.

UNetbootin
When creating boot media and live discs, you might also consider using USB storage for it. Having an easily portable, live-booting USB stick can be very useful for fixing problems, such as boot issues or hardware and driver errors, as it doesn't need to actually load up into the installed Linux distro.

Wireshark
Network troubles might be quite far-reaching in your network, and Wireshark is an excellent piece of software for going through your network and seeing where problems and faults may be arising locally. It works on individual machines and servers and includes a web frontend for ease of use.

Bacula
Backing up is an important thing to do and a safety net if you're having irreparable problems on your system. Bacula is one of the best pieces of software for this, as it includes comprehensive backup solutions with included scheduling and networking features.

COMMON PROBLEMS

Routine solutions to solve many widely-faced issues

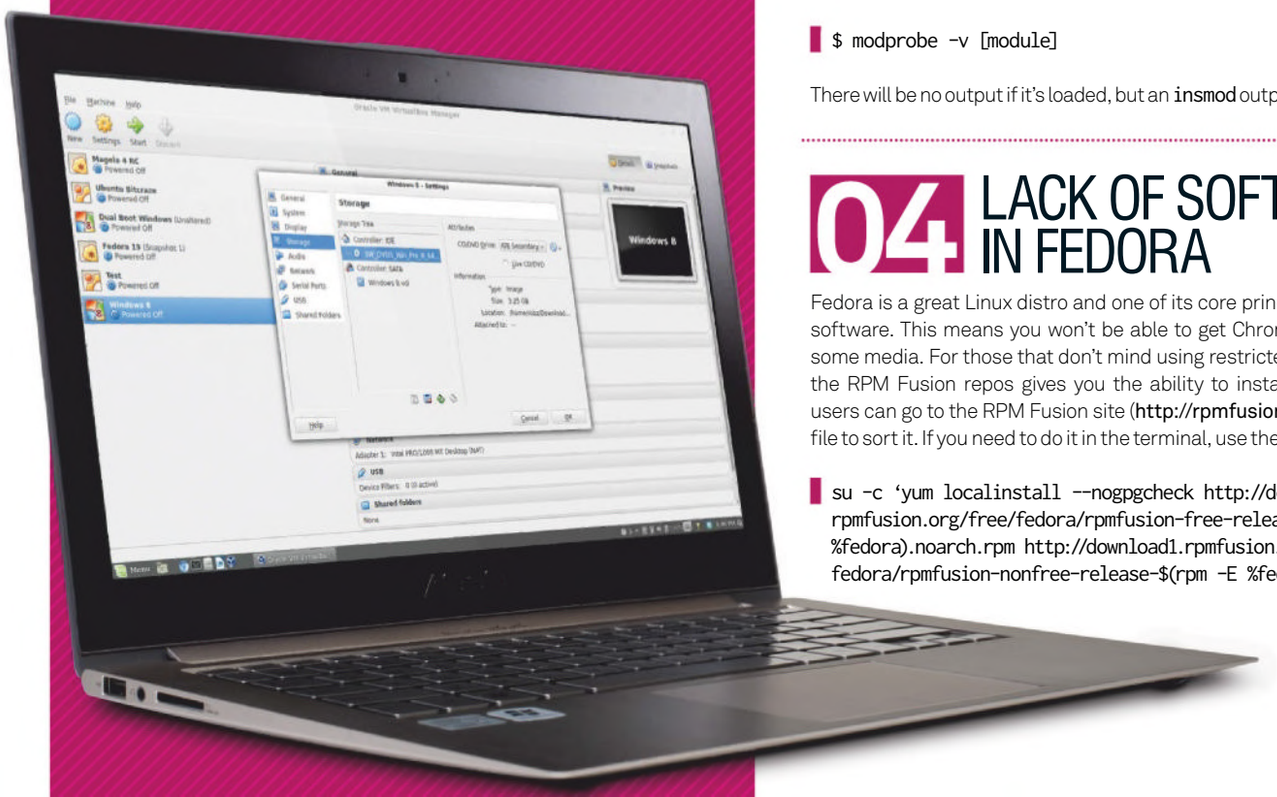
01 SWITCHING FROM WINDOWS

Moving to Linux from Windows can be a hurdle to overcome for a lot of people. Even if you just need to advise someone, it's a good thing to know the ways to make the transition as smooth as possible.

First of all, assuring someone that their files, documents, music and such will work on Linux is a good step. If they are using Microsoft-own software like Office and Media Player for example, LibreOffice and the myriad of media software on Linux can play it just fine. For a lot of people, this transition just requires backing up the files before making the jump.

For software that does not have a Linux equivalent or requires the Linux version, using Wine may be an option. While it's not guaranteed to work, it may be something that is worth looking into. A more surefire way is to create a Windows virtual machine on the system that can be accessed whenever the software is needed – it's not a perfect solution but it can be better for some than just using Windows all the time. So the initial jump isn't as daunting as you first thought.

"It's not a perfect solution but it can be better for some"



02 FILE AND FOLDER PERMISSIONS

Sometimes you hit a problem where you can't create, delete or edit files without using a root account or **sudo**. While this is fine for working in the terminal, when you're trying to run scripts it can be an issue. If you've had to create folders and files using **sudo** when they don't require root-only access, the best option is to change permissions. For files and folders it's the same: you'll use **chmod** along with a string of numbers to indicate what to change it to, for example:

```
$ sudo chmod directory 666
```

... which will make everyone able to read and write in the directory. 777 on a file will let anyone execute it too. The first number applies to the owner of the file, the second number to the group the owner's in and the third number is anyone.

03 HARDWARE PROBLEMS

Not every piece of hardware will automatically work on Linux, but usually it's just because you need to turn something on. Using **sudo lspci** and **sudo lsusb**, you can list the hardware devices connected to your system. Each device will have an address code, something like 01:00.1 for **lspci** – note down the address for the problem device and use **sudo lspci/lsusb -s [code] -v** for more info.

This should tell you which module has been loaded for the device. It's probably not the right one, so do this again while using a live disc and check to see which module it uses. If it's different, go to your installed system and use:

```
$ modprobe -v [module]
```

There will be no output if it's loaded, but an **insmod** output if it's just loaded it up.

04 LACK OF SOFTWARE IN FEDORA

Fedora is a great Linux distro and one of its core principles is to only use free software. This means you won't be able to get Chromium or be able to play some media. For those that don't mind using restricted or non-free software, the RPM Fusion repos gives you the ability to install more software. Most users can go to the RPM Fusion site (**http://rpmfusion.org**) and download the file to sort it. If you need to do it in the terminal, use the following:

```
su -c 'yum localinstall --nogpgcheck http://download1.
rpmfusion.org/free/fedora/rpmfusion-free-release-$(rpm -E
%fedora).noarch.rpm http://download1.rpmfusion.org/nonfree/
fedora/rpmfusion-nonfree-release-$(rpm -E %fedora).noarch.rpm'
```

05 NEED TO KILL A TASK

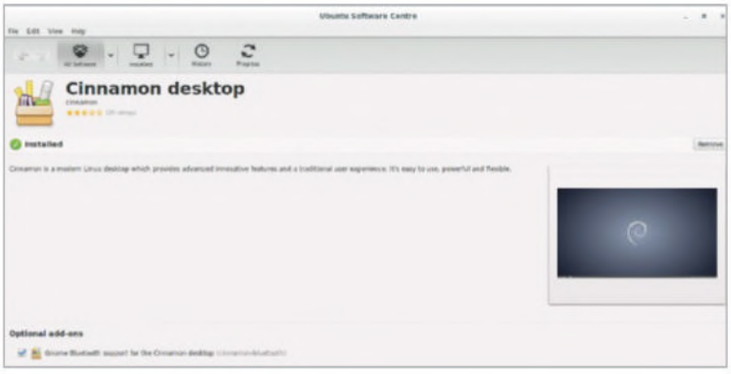
A program might be running and taking up a lot of resources, and sometimes you just can't turn it off the normal way. This calls for use of the `kill` command in the terminal – first find out what the PID of the software is in **top**. Simply run the **top** command in the terminal and it will show up – press Q to quit out and then:

```
$ sudo kill [PID]
```

06 ACCESS LOGS TO VIEW ERRORS

A good way to figure out a problem is to access the log files on your system to see if an error message may have been spat out. This might be in a major log file or one specific to the software, and they can be found in the /var/log folder – either navigate there in a file manager or look them up in the terminal.

"A good way to figure out a problem is to access the log files to see if an error message may have been spat out"



07 CAN'T FIND CINNAMON DESKTOP

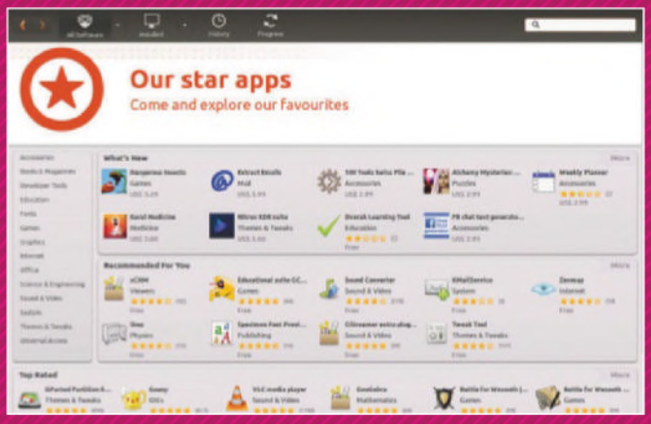
The Cinnamon desktop is becoming a lot more popular but it's not quite everywhere yet – the most noticeable omission is from the repos of Ubuntu. There's also now no officially maintained repository for the Ubuntu build of Cinnamon anymore, but there is a third-party one you can use. Open up a terminal and add the PPA with:

```
$ sudo add-apt repository ppa:lestcape/cinnamon
```

Now install by doing an `apt-get update` and `apt-get install cinnamon`.

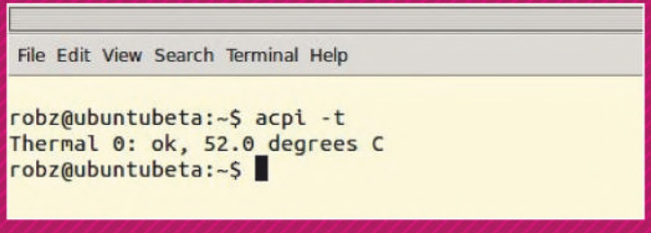
08 KEYBOARD IS IN THE WRONG LANGUAGE

Sometimes during installation, the installer won't properly set the right keyboard, or maybe you've got a new layout and it's not working properly. This can usually be changed in the settings, but in LXDE you can set multiple keyboards and switch between them with Ctrl+Space.



09 IDENTIFYING YOUR PACKAGE MANAGER

Not every version of Linux uses apt or yum in the command line to install and manage packages. You should be able to identify which package manager your distro is using by what its base is: Fedora and Red Hat distros use the yum package manager. Debian and Ubuntu-based distros use apt and dpkg. OpenSUSE uses a mixture of YaST, zypper and pacman to handle package management. Gentoo uses portage and the `emerge` command and Arch uses pacman. The latter two do a bit more compiling than the other distros, so sometimes can take a little longer for software to install.



10 CPU TEMPERATURE

If you think your system is running a bit hot, you can check the temperature to see if and when it's getting a too toasty. For this we need to install ACPI, so open up the terminal and use:

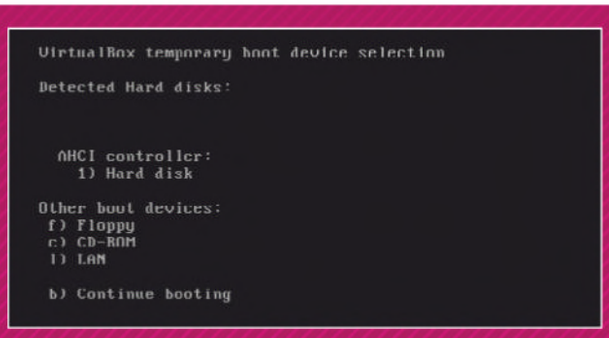
```
$ sudo apt-get install acpi
```

Then you can check the temperature with:

```
$ acpi -t
```

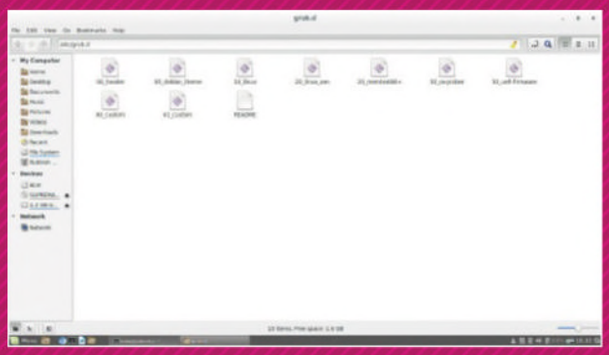
BOOT ISSUES

Get GRUB working and boot back into Linux



11 BOOT LOCATION

If your system won't boot from a CD or USB stick when you turn it on, look out for a boot menu option during the first screens. It's usually tied to an F key but failing that, boot into the BIOS and change the boot order selection so that your hard drive is lower down in the process.



12 BOOT AFTER WINDOWS INSTALL

Windows won't let you boot into Linux, unlike GRUB, which enables you to do both. After a dual-boot install, you'll have to restore GRUB like in Problem 16 (just to the right). Then go one step further and do an **update-grub** in the terminal, followed by **grub-install /dev/sda** to boot to Windows too.

13 CHANGE DEFAULT GRUB OPTION

There are two ways to do this – if you're just using two operating systems, you can change the name of the OS-prober file in /etc/grub.d to be before the main Linux file and perform an **update-grub**, or you can change the default selection number in /boot/grub/grub.cfg.

14 REMOVE SPLASH SCREEN

One way to find out more niche issues is to see what happens during boot – but it's hidden with the splash screen. In GRUB, you can turn off the splash screen by going to /etc/default/GRUB and removing the "quiet splash" answer on the GRUB_CMDLINE_DEFAULT_LINUX option.

15 ADD OTHER BOOT OPTIONS

The beauty of GRUB is that you can add boot options to it – either different ways to boot your distro, or adding other distros if you multi-boot. For multi-booting, the quick way is to use the pre-installed os-prober (operating system prober) in GRUB. To do this, all you need to do is run an **update-grub** for it to check what else is installed and add to the grub.cfg. Otherwise, you can create custom installation option files in /etc/grub.d, named in order of when you want them to boot.

16 GRUB CANNOT BE FOUND

This one doesn't often happen just out of the blue – usually there's a reason, like you've installed another distro to your system, for example. But we've turned on a laptop once or twice and it's just… happened. Either way, don't panic as you can easily restore GRUB.

First of all, get a live-booting medium of your preferred distro – Ubuntu will do, but so will Fedora on either CD or Live USB. Stick it into the problematic machine and restart it to load up the live environment.

Once you're in the desktop, mount the hard drive where your distro is installed to in the file manager and then open the terminal. Make sure GRUB is installed by trying to install the GRUB package (for Ubuntu, **apt-get install grub**; Fedora is **yum install grub**, etc). Still in the terminal, use **fdisk -l** to figure out the location of your hard drive (such as /dev/sda). When you have that you can type:

```
$ grub-install –root-directory=[Mount point of hard drive] /dev/sda
```

The mount point being where the file manager mounted the filesystem and not /dev/sda. You should then be able to reboot and go back into your original distro.

17 NEED TO MANUAL BOOT

This is not an easy one – you may be given a boot prompt at GRUB asking you to manually type in the boot command. If you can't reinstall GRUB for some reason, or don't want to, you'll have to know the following pieces of information: which version of the Linux kernel you have and the UUID of the hard drive. These can be quite long strings, so you'll need to write them down. Once you have though, you can boot by typing something like:

```
$ linux  /boot/vmlinuz-[kernel version]-generic root=UUID=[UUID of drive] ro  quiet splash
```

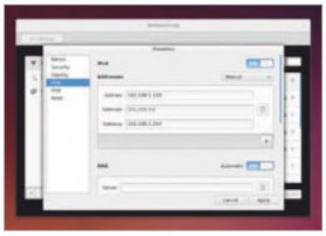
… and …

```
$ initrd  /boot/initrd.img-[kernel version]-generic
```

NETWORKING FIXES

Tips to help fix your network issues and create stabler LAN

18 SET A STATIC IP



While a router will tend to dole out DHCP addresses to the same network interfaces every time, it's not consistent. When you have port forwards active or very specific connection addresses set up around your home network, it can be best to set a static IP.

In desktop environments there will be a graphical network manager which will let you set manual settings for IP, gateway, subnet mask and DNS. Otherwise, you can open /etc/network/interfaces using **nano** and add it manually.

19 TEST YOUR WEB ACCESS

If you're having trouble loading a page or two, you may be struggling to connect to the Internet. Or the server you're trying to contact may be having problems. To help try and differentiate between a bad net connection and a busy server, you can use the **ping** command to see if communications are getting through properly. Open up the terminal and ping Google with:

```
$ ping www.google.com
```

It will send small packets to Google asking for a response and record the response time. It's also worth trying a different website if Google is something you're having problem with.

20 QUERY DNS VERSION

If you're browsing on the Internet and some websites seems to be fine while others simply aren't, there could be any number of things causing it – most of them not your fault. However, one issue that will catch some people out is whether or not their DNS is up-to-date. To check what DNS you're using, you can use **dig** to find more details. Open up the terminal and type:

```
$ dig www.google.com
```

And look at the answer section for the DNS addresses.

21 CHECK YOUR NETWORK ACCESS

Similar to Tip 19, when trying to diagnose where your Internet or network is failing, it can often be a good idea to check to see if your system can talk to the router at all.

You'll likely know if this is the case from connection reports, but if you're still struggling to find a problem and perhaps your router has connected your system using a weird address, it can be a useful step. To do it, open up the terminal and use:

```
$ ping [Router IP]
```

"To check what DNS you're using, you can use dig to find more details"

22 SETTING A DNS

Somewhat related to Tip 20, if your DNS is playing up and you need to change it, adding a different DNS in the settings might be one way to help you. There are a couple of free and public DNS servers, such as Google's Public DNS, available to use. To change them on your system, you need to open the terminal and edit the resolve.conf file using **nano**. Change the nameserver value and add a second value as well, to complete the process.

23 IF ALL ELSE FAILS, TRACEROUTE

If you still have no idea what is going on in your network then a deeper scan of what exactly a package is going through might help you track down a problem. You can do this with **traceroute**. There are a huge amount of options you can use for this, but you can easily get a picture of the process by using the following command in the terminal:

```
$ traceroute www.google.com
```

24 MANUAL WI-FI SETTINGS

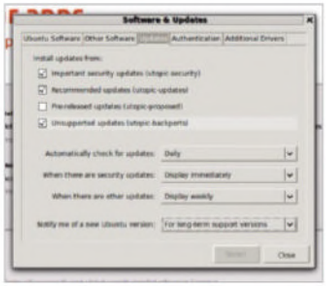
If using wireless in the command line, setup can be tricky as you must account for more than a wired connection. These settings will go in the interfaces file along with wired information, found at /etc/network/interfaces. Here's an example setup:

```
auto wlan0
iface wlan0 inet dhcp
        wpa-essid [SSID of network]
        wpa-psk [Password]
```

INSTALLATION ERRORS

From installing libraries to software, there can always be a problem

25 UBUNTU LTS IS NOT UPGRADING



If you're using an Ubuntu LTS just because it's the freshest version that you installed, but you would prefer to keep up with the latest releases, you may find that it won't tell you when these latest releases actually come out. This is because by default, an Ubuntu LTS will only update when another LTS is officially released.

To change this, open up the Software Centre, go to Edit and then Software Sources. Under the Updates tab, there's an option to change when you are notified of a new version of Linux. Change it to 'Any new version' and next time it does a periodic check, it will let you upgrade straight away. No more being stuck in a version of Ubuntu that you don't want!

26 REQUIRE BUILD TOOLS

When compiling software, your distro needs the right software installed to do this. Packages like the GCC compiler and related tools are not always installed by default any more due to the convenience of package managers and downloadable binaries. Not every distro lets you install it the same way though.

In Debian and Ubuntu-based distros, you can install it with the package build-essential, which as the name implies comes with the essential packages for actually compiling and building software. In Fedora, it's slightly different as you have to install a group of packages using **yum**, as shown below:

```
$ yum groupinstall "Development Tools"
```

27 NEED TO ROLLBACK A PACKAGE

While updating is usually a great idea to keep ahead of bug fixes and security updates, every now and then there may be a problem with the update either for everyone or specifically for you. Not every package manager has the ability to rollback specifically, but there's a way you can do it. Apt requires you to first uninstall the package, use **apt-cache showpkg [package]** to get the available versions and then install it with **apt-get install package=version**. In Fedora, you can just use **yum downgrade [package]-[version]** to rollback.

28 MISSING DEPENDENCIES

When installing software, the required dependencies often aren't met. This means that libraries and packages required aren't available. Generally, you can fix this by looking at the output and seeing which package you need to install. In Debian-based distros you can install **auto-apt**, which will put the compiler on hold when it finds a missing dependency so it can install from sources. To run it with **auto-apt**, configure with:

```
$ sudo auto-apt ./configure
```

"Now and then there may be a problem with the update"

29 UPGRADE A CD IMAGE

When testing daily builds of a distro, there's often no way to update software. But ZsyncCdImage uses **zsync**, software that lets you selectively update with the files that have changed to enact an ISO-wide update by changing the files on the image. Install **zsync** and update with:

```
$ zsync [URL to daily image]
```

Then burn a new disc and reinstall.

30 CAN'T INSTALL POST-INSTALL

This problem occurs with distros based on Mandriva and PCLinuxOS – you'll install the distro but can't add software. The distros are looking for software, so if you're trying to install a package that's not present, you'll be in trouble.

To fix this, go to Software Management and you can configure media sources. Choose the standard mirrors and deselect the CD or DVD as the source. Update the repos by reloading the software manager and you'll be good to go.

31 REMOVE REPOSITORIES

You may need to do this for many reasons: the software a third-party repo is distributing is broken, the repo is obsolete or maybe it's never worked. Removing it is different in various distros.

In Debian-based distros, remove a repo by editing the /etc/apt/sources.list to delete the details of the repos. If you're using Ubuntu and a PPA, you can use **apt-remove-repository [ppa name]**. In Fedora, repos are contained in labelled files in /etc/yum.repos.d, so delete from there.

32 INSTALL FREEZING OR BUSY

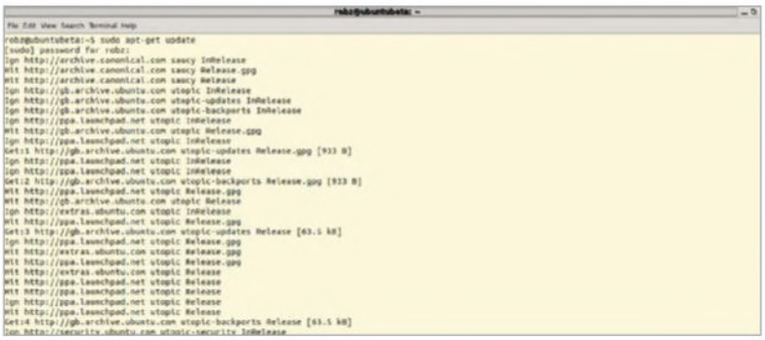
When something like this happens, one of the primary suspects is an errant installation or update program going on while you're trying to install. You'll need to look through logs and your system to see what may be using the installation tools, then either wait for it to finish or kill it off so that you can continue.

33 CHECK DOWNLOADED FILES

Sometimes installation issues can be caused by incomplete, corrupted or incorrect downloads. This is why a lot of FOSS comes with an MD5 hash associated with it, which you can check against the file itself to make sure it's correct. To do this, open the terminal and type:

```
$ md5sum [filename]
```

And compare it with the hash sum given online.



34 PACKAGE NOT FOUND

If you've added or activated a repository recently and are having trouble finding it, or there's new software in an existing repository that's not showing up, your software lists might not have been properly updated yet. In Ubuntu, you can update it from the terminal with **apt-get update**, and Fedora has **yum update**, although the latter will also enact any software updates at the same time.

35 NEED TO FORCE AN UPDATE

While getting dependencies sorted out is one way to make sure software installs, other times just forcing the package to install may fix a problem. While this may be a solution for dependency problems, there could be other issues blocking an install that don't matter for you. In Ubuntu you should download the package using **apt-get download [package]** and then install with **dpkg -i [package]**. In Fedora you can use **rpm -ivh –force [package]** to do this.

36 DISTRIBUTION UPGRADE

If you don't have the benefit of a graphical package manager or software manager to perform a distribution upgrade with, Debian-based distros can make use of **apt-get** to do this. In the terminal, do an **apt-get update** and then follow that up with:

```
$ sudo apt-get dist-upgrade
```

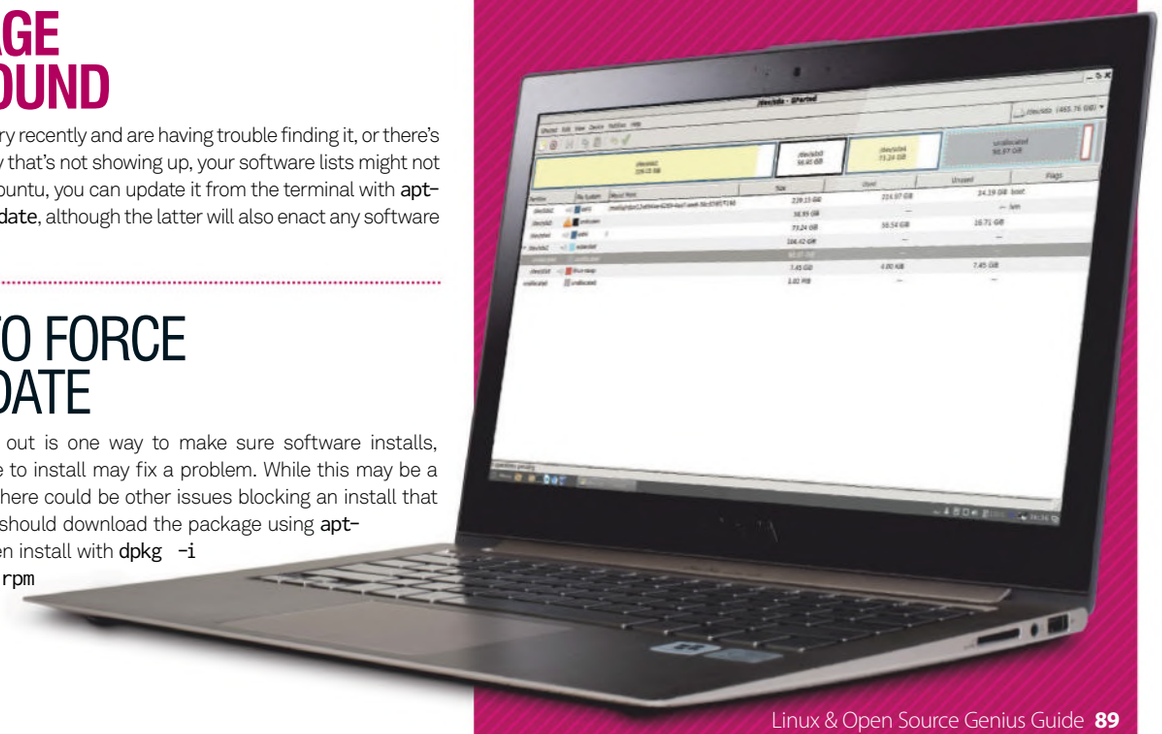
37 FRESH INSTALL STILL NOT WORKING

If your new install is still failing itself, then a good course of action is to just try and install the system again. Rewrite the live medium to make sure it was created correctly in the first place and do a complete install; it's rare, but sometimes something will have gone wrong in the process and just trying again may create different results.

38 PARTITION PROBLEMS

Some distros have some very easy and straightforward partition options, while others very much don't. What these other distros have in common, though, is that they'll let you mount to existing partitions. Using a live disc that contains GParted (which is most of them), you can set up the partitions as you want much more easily. Or just use **fdisk** in a terminal.

"Or just use fdisk in a terminal"



39 STOP A SOFTWARE RAID

If you need to stop an mdadm software RAID for whatever reason, first make sure you take any files you want off the device. Once that's done, and assuming yours is md0 with drives sda and sdb, you can disable it with:

```
$ mdadm  --stop /dev/md0
$ mdadm --remove /dev/md0
$ mdadm --zero-superblock /dev/sda
$ sfdisk -d /dev/sda | sfdisk /dev/sdb
```

40 HARD DRIVES FILLING UP

If you think your hard drive is getting full a bit prematurely then you should check some of the temporary files. Empty any trashcan-style location you have and give the cache of your browsers a clean. Have a snoop around the home folder for anything you may have missed too.

41 NEED TO CREATE A PARTITION TABLE

This is usually something that affects hard drives that have been in many different systems rather than new ones, but sometimes you will need to create a fresh partition table. The easiest way to do this is to load up GParted and add it manually in Device>Create Partition Table.

42 FRAGMENTED LINUX DRIVE

Ext filesystems that Linux uses, and Linux itself, have ways of keeping files from fragmenting. However, it's not perfect so fragmentation may occur eventually. There aren't many defragmentation utilities for Linux, but defragfs (**http://sourceforge.net/projects/defragfs**) will do the job well enough if you're worried about fragmentation.

HDD FIXES

There are a few things you may need to do to maintain a hard drive

43 RECOVER YOUR DATA

One of the biggest fears of computing can be losing all your important data – that's why there are so many backup solutions available at varying levels of safety and security. Sometimes you just have to prepare for the worst, and being able to recover data from a hard drive may be needed eventually.

TestDisk (**www.cgsecurity.org/wiki/TestDisk**) has functions to undelete files and recover partitions. It's slightly better at the latter due to the nature of the recovery, and all you need to do is load it up on the problematic system (even on an installed distro, if you can) by moving to the folder and typing **sudo testdisk-static** in the terminal.

From here you can create a log and then select a disk to try and rescue. After a scan, it will try and determine the partition table type and you can confirm or change it if you know it's supposed to be different. Then you can Analyse to see if there's a partition error, do a quick search to find any missing ones and Write to restore the partition.

For undeleting files, you can choose the Advanced option instead of Anaylse, select a partition to work on and it will undelete all of the deleted files and move them to a specified location.

44 ADD MORE SPACE

There are a few ways you can do this. One of the easiest and lowest maintenance methods is to add, format and mount a new hard drive into your home directory or wherever it's needed. This way, all the original drives are untouched and you know it will still run just fine from there. Otherwise, you can clone the hard drive using Clonezilla (**http://clonezilla.org**) and restore it to a newer, bigger hard drive and then extend the Linux partitions into the available space.

45 OUT OF VIRTUAL SPACE

If you've set up a virtual machine in VirtualBox for testing, it's not uncommon if you're using it often enough to run out of virtual space, even if you gave yourself a fair amount that was dynamically allocated. You cannot exactly extend into your host hard drive space, but you can always create more space to use. Go to Settings>Storage and add a new hard drive to the SATA controller. Boot into your VM and format the device, then mount it to a necessary directory to increase storage.

GRAPHICAL PROBLEMS

Sort out your drivers and modules, and learn how to exit X

46 VIRTUAL DISPLAY

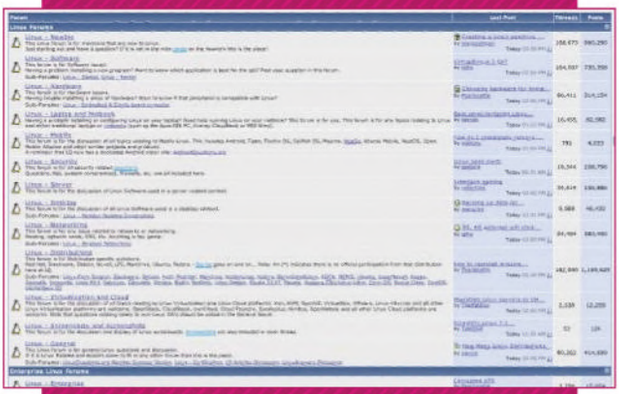
Often when you set up a virtual machine in VirtualBox, you will find that the display does not completely fill up your screen. While this cannot be changed on live discs, if you've installed and configured a virtual machine then you can add kernel modules that enable you to use a better resolution. In VirtualBox, go to the Devices menu and click Insert Guest Additions. It will download an ISO that you can use to install the new kernel modules and then use the full-screen resolution as you wish.

47 INSTALLING DRIVERS

Graphics drivers on Linux come in two forms – restricted drivers that are downloadable or come with a distro such as Ubuntu, and reverse-engineered drivers such as Nouveau. Usually, if you're having some form of graphics issue, the first thing you can try and do is switch between these two drivers. To download the proprietary ones, head to the AMD, NVIDIA or other website and either use the auto-detection tool or select the driver manually. They go through an automated installation method, which is a good first step to fix graphical issues.

LINUX ANSWERS FOR LINUX QUESTIONS

The problems we've covered in this article are ones that we feel are fairly common and universal to all forms of Linux. However, that won't stop you coming across an issue that is much more specific to your setup. While your first line of defence should be a few carefully worded Google searches, if that's coming up with nothing then your best bet is to check out Linux questions at **LinuxQuestions.org**, which has an excellent community of very clever and helpful folk who can help point you towards answers to your problems. Hopefully, though, it will never come to that.



Above Search through the forums before posting a new question, in case yours has already been answered

48 GET TO THE COMMAND LINE

Sometimes X, the graphical display server, will freeze and you might be unable to do anything as a result. Additionally, it could become horrifically slow or have another error coming out of it. Instead of performing a hard reset or simply waiting it out, you can always switch to the command line and try to fix things from there. To do that, just hit Ctrl+Alt+F2 and log in with your normal username and password to access all of the various command line tools.

49 RETURN TO DESKTOP

If you have had to go to the command line for some reason, as we just advised in Tip 48, you can also return to the original graphical desktop using Ctrl+Alt+F7 – there are actually instances that can run on F3 to F6 as well, which are usually command lines too. You can quit X on the F7 instance and start it up on another one wherever you wish, so if F7 doesn't get back to it, try the other F keys instead to get a result. The choice is up to you.

50 HOW TO RESTART X

If you need to turn X off and on again, try using Ctrl+Alt+Backspace to restart X. If that doesn't work, use Tip 48 to go into the command line instance and kill the display manager using:

```
$ sudo service [display manager] stop
```

Your display manager could be LightDM or GDM – you may need to google it. Once it's killed, use `sudo service [display manager] start` or `startx`.

TRIPLE BOOT

Two operating systems are so last year – here's how to start using three of them to maximise your productivity

Dual booting is a staple of being a Linux user these days. Classically, a lot of people think of this as Linux and Windows coexisting together on one machine. There are people who just have two Linux distros though and there are many reasons why this could be the case – testing on two systems, one for leisure and one for work, is just one example of why two can be useful.

We can easily take a step beyond that though. It's doable (storage space permitting) to have three operating systems residing side-by-side. Whether you're having two Linux distros and a Windows install or simply three Linux distros, the concept is quite similar to dual booting and a natural extension of the practice.

Over the course of this feature, we will teach you how to perfectly partition your system, from a fresh hard drive to a pre-existing install, as well as a few tips on the best methods of installing the systems to get them to work together.

Why go for triple boot?

As we noted above, there are many reasons why you would want to both dual boot and triple boot, and they depend entirely on how you use your computer and how often you need to use different environments for different tasks.

One reason is often Windows – however we feel about it, many of us need it in our day-to-day lives. It could be something as simple as enjoying playing new games, which aren't always supported on Linux, or it could be the case that you are a designer who needs to use the industry standard Photoshop or InDesign. You can even install OS X for a Hackintosh build if that's more to your taste.

A key reason to further extend a dual boot setup is to preserve your main distro – the one containing the bulk, if not all, of your personal data and media. There are innumerable reasons as to why you may want or need to use different distros on a regular basis, and sometimes live-booting or virtualising just doesn't cut it – in such cases, it is incredibly convenient to have a third partition onto which you can install the distro you temporarily need to use. Non-Linux OSs aside, it could be something like wanting to have, for example, a Pentoo partition for testing alongside your main Debian distro, with a third slot for distro-hopping. It really is down to you!

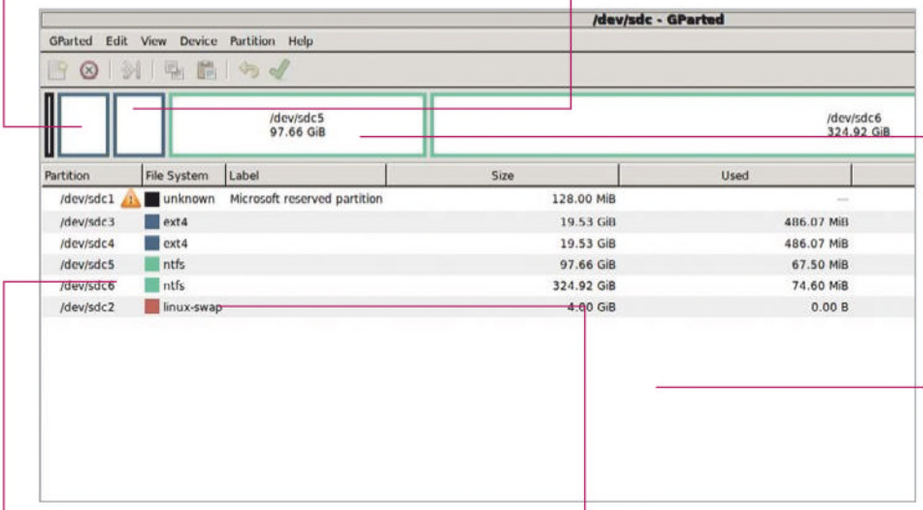
Partitioning

Set up your hard drive so that space is usefully split between your different distros

In the first partition we have our primary default distro – this could be your work or leisure distro. We recommend about 20 GB ext4 partition for a Linux distro

The same as the first partition – a 20 GB ext4 partition for your other version of Linux. If you're triple booting Linux, you can use a third one of these quite easily

This partition is larger as we will install Windows here. Make it NTFS so Windows can see it during installation. It's best to give Windows 100 GB of space to be safe



We'll create a shared storage partition for all operating systems. This can be made up of all the remaining space and it's best to keep it to NTFS so everything can use it

This is the swap partition, used with the RAM when Linux is running. Similar to Windows' page file system but that resides on the main Windows installation partition

If you plan to set up the hard drive and install the operating systems from scratch, it's best to use GParted – found on most live distros or any maintenance distribution

Above is a useful setup for triple booting your system, but this is only a guideline. The 20 GB sizes for the Linux distros take into account just purely installing packages – in many ways it's a very liberal estimation of how much space you're going to use, however this depends on your development habits and what kind of software you are planning on using. The order is also fairly arbitrary – it won't make any difference to disc speed but it may make sense to you personally.

While we do recommend a shared storage partition, the file structure of the home folder in Linux and Windows is quite different, which can easily complicate things.

Windows and Linux both allow you to mount specific parts of the partition to specific locations in their hierarchy though, which can make it a lot easier and quicker to organise. However, another option that you can think about is splitting up the storage partition between the two.

Installation order

Windows first, primary last

Installing Windows first can actually make the installation process a lot smoother – this is also good news if you've already got an existing or new Windows computer. The major benefits of installing Windows first is that you don't have to mess around with recovering and rebuilding GRUB at the end of the installation process, and it won't try and overwrite your currently installed Linux distros during its own installation process. The GRUB benefits also apply to installing your primary distro last, as you'll then be able to easily modify and update it later.

Linux first, Windows second

Performing installation in this order has its advantages by more easily tracking what you're installing and where. If you're setting up a disc from scratch or already have a Linux distro installed, you can use GParted straight away to get the disc formatted to your specifications. This means that while you're still in the live disc you can do the first installation. This can save a lot of time if you have limited resources for creating live discs or live USBs – you're already in Linux to edit the partitions, so why not install it?

"This order has its advantages by more easily tracking what you're installing and where"

Installing Linux

Installing Linux alongside another distribution can be easy with the right distro

Installing Linux has been reduced down to the bare minimum interactions these days for a lot of distros. Ubuntu, Fedora, openSUSE and all your major, modern distros have their own graphical wizard either shared between them or created for the distro. Usually it's a case of just hitting install and overwriting the disk, but if the hard drive partitions are set up as we suggested then you won't want to do that.

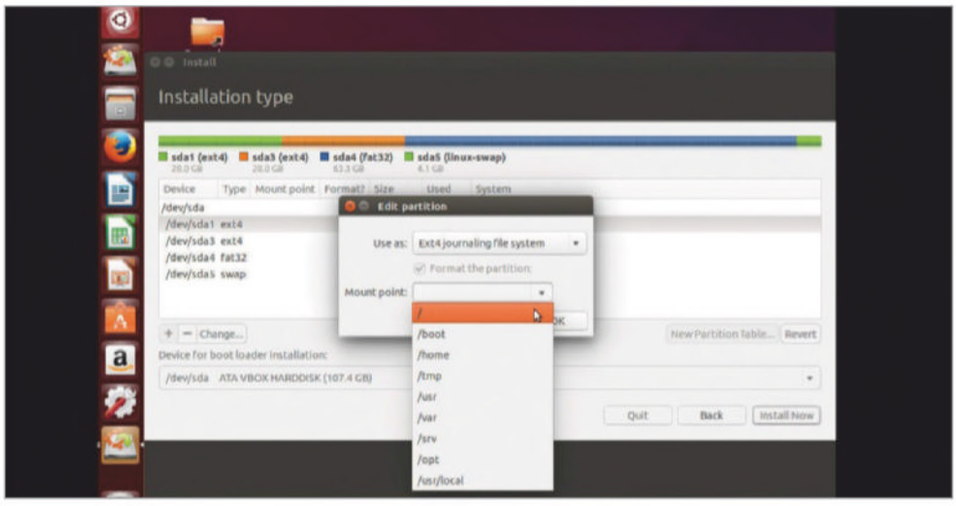
In these distros though, you will be able to do one of two things: install alongside or specifically set up a place to install the root file system. The root file system is the core of the distro, where all the files to run it are stored, and it is represented by a '/'. Installing some distros from scratch will make it create a separate partition for the boot files or home folder and you can certainly make your custom partition setup do that as well.

On less advanced distros – perhaps those designed for older systems – the installation process can be a lot more involved. It will require you to know what partitions you've got and where they live on your system. While they may have their own partition software it is likely to be a lot more manual, so in this case we'd still recommend using GParted with another live distro first to get the partitions sorted beforehand. During this process make a note of what the hard drives are called during the partitioning process – this will be something like / dev/sda for the hard drive and sda1, sda2, sda3, etc for the individual partitions. These numbers won't always be in the order you expect though, so it's best not to guess it.

■ Always read the disk options during installation – it's easy to erase the disk entirely

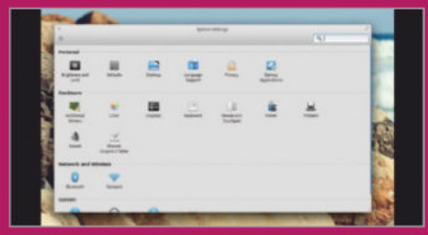
All distros, live and installed, will automatically use any swap partition on the system. You only need one of these for your system as you are just running one distro at a time, and you don't need to set it as the correct swap when installing either.

For further installation advice always make sure to read the available options and if all else fails, seek out original documentation on the distro's website.



■ In Ubuntu you can select custom partitions for different areas of the distro's files

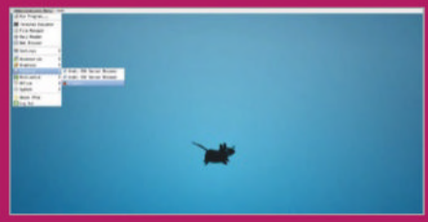
Types of distros



Everyday

elementary OS

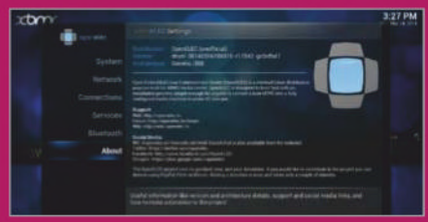
A beautiful looking distro that is easy to use and yet still offers everything you would want Linux for.



Development

Arch Linux

A serious distro for serious Linux users, Arch has everything that you need to develop your skills.

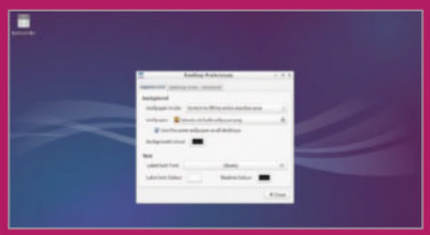


Entertainment

OpenELEC

Fully explore and experiment with Kodi using OpenELEC. It is the best way to view media on Linux.

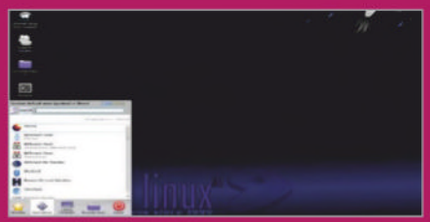
Choosing distros to use is easier if you think in terms of their categories



Lightweight

Lubuntu

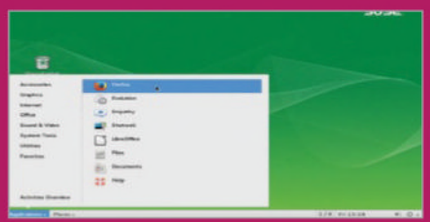
Being light without sacrificing anything is one of Lubuntu's major pros. There are many more advantages though.



Rolling release

Gentoo

You can get the very latest packages and updates in one of the most customisable setups around.



Working desktop

openSUSE

It's great for enterprise and it's also great if you want to just do some work without any distractions.

Recovering GRUB

Get the boot menu back if things go wrong or manage it with a different distro

One of the issues you may find with installing one or two distros next to each other is that you might mess up GRUB, the boot manager used by Linux to actually boot into the distros and other operating systems. You may also want to manage it with your 'default' OS. Both of these can be fixed using our recovering GRUB guide below but unfortunately this won't help you recover an operating system you've written over.

01 Live boot

Any of the distros we've been using will work for this – you can even technically do it from another installed Linux distro if you're already inside it. When you boot into the live Linux, you may need to install the grub package. Ensure that you do it from the terminal before continuing.

02 Mount the hard drive

Some distros like Ubuntu, for example, will let you click and just auto-mount the internal hard drive from the live environment. However, you can also do this in the terminal. Mount the primary install partition to a logical spot using something like:

```
$ mount /dev/sda1 /mnt/
```

03 Restore GRUB

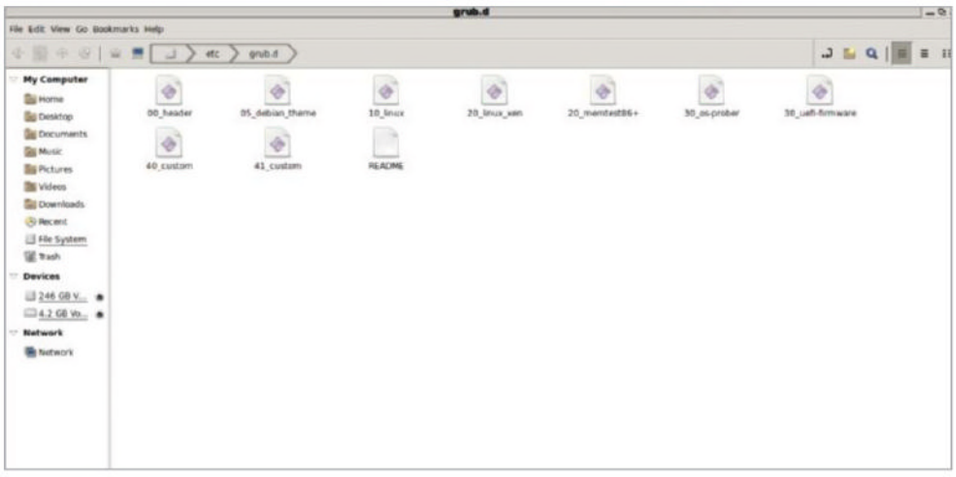
Assuming your primary partition is sda1, your installation hard drive is sda and you mounted it to mnt, you can now restore GRUB using the data from the partition that you just mounted with:

```
$ grub-install --root-directory=/mnt /dev/sda
```

04 Reboot

After a reboot, GRUB should be back to normal and at the very least you can boot into your main distro. From there you may need to update GRUB further – to do this, open the terminal and perform the following two commands:

```
$ update-grub
$ sudo grub-install
```

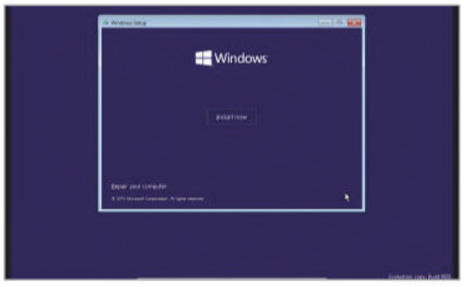


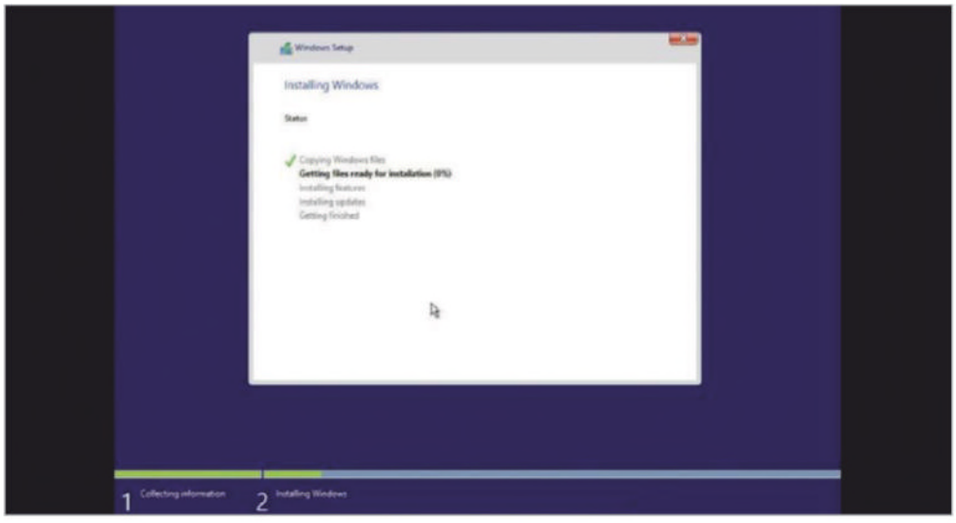
■ Your GRUB components all live inside the etc folder of your Linux distro

Windows installation

Microsoft's operating system is a necessary evil to some, but here's how to summon it safely from its dark pit

If you've already got Windows installed, we suggest looking over the page to figure out the best way to prepare for installing other operating systems alongside it. Otherwise, installing Windows when Linux is already there is not quite as easy as its Linux counterparts. Windows would prefer to completely wipe the disc and set itself up as the ruler of your computer, but with some persuasion you can get it to play nice.

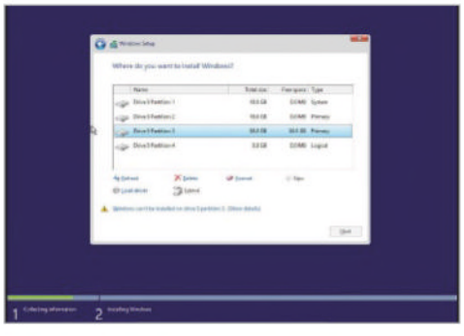




■ Feeling adventurous? Grab and burn the Windows 10 Technical Preview from **bit.ly/1y8MoE2**

01 Prepare to install

Put the disc in and boot your PC. The first step in installing is to set the language as you normally would. Click install and then agree to the licence. After this, you'll be asked how you want to install; click on the Custom install option to install from scratch.

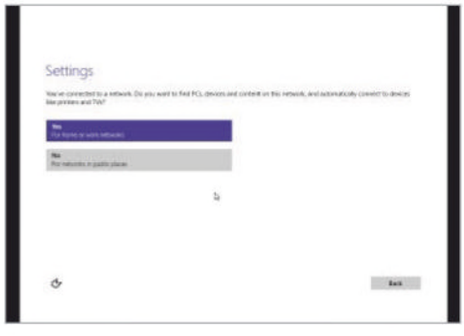


02 Storage

Select the empty partition we created before in GParted as the place to install Windows to. It will recognise it as NTFS, with the Linux partition as unknown – it will also completely reformat this partition once again, so make sure there's nothing on there.

03 Wait a while

Windows can take a while to install and will go through several phases, including rebooting once or twice during the process. Leave it alone and it will do its thing without any interruptions. Now is a great time to go and make yourself a cup of tea.

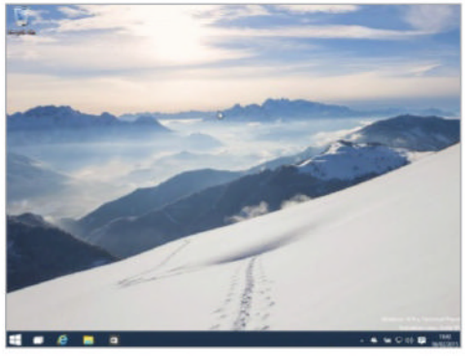


04 Personalise your install

On Windows 10 (or Windows 8.1, if you're playing it safe), you'll now need to make some basic settings for your data and the way that your system should work. Go through the wizard with your own choice of settings before entering in your account details.

05 Access a microsoft account

You'll need to log into or set up a Microsoft-based account in order to use the latest versions of Windows. If you already have one then you can enter it here and log in. Otherwise, you'll need to link an email address to a new account.



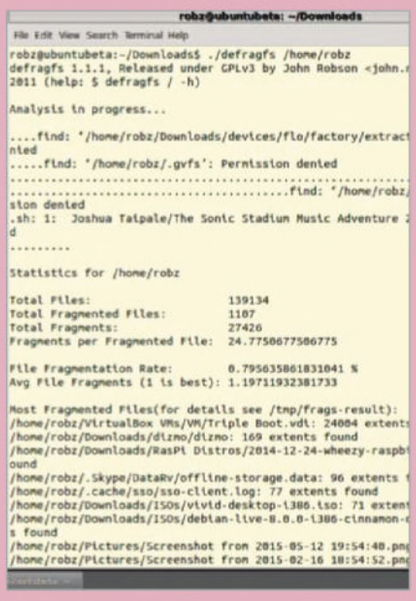
06 Final setup

Wait a while and the wizard will grab your account settings and any other data you may have associated with a Windows 10 installation under your Microsoft account. After this, it will bring up the desktop and allow you to start using it.

"You need to be careful when shrinking the partition"

Defragging Linux

While ext3 and ext4 are much more efficient, they can still exhibit fragmentation. Shrinking of the partition might still corrupt files if parts of them are located towards the end of the disc, although it's much more unlikely. If you're shrinking right down to the line, you're asking for trouble. Linux defragmenters are few and far between though (however defragfs is worth a try: sourceforge.net/projects/defragfs), so one of the best methods is to temporarily move all of your personal files onto portable storage, while setting up the partitions and such, and be conservative. Again it is worth mentioning to always back up important files on these kinds of tutorials anyway.



■ Defragfs gives you quite detailed and useful output in the terminal

Shrink partitions

The NTFS file system isn't suited to editing its space use but there are ways around it

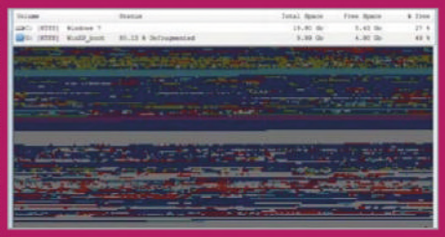
If you're starting with Windows already installed onto your system, then you'll need to start making space so that you can install your other distros. This means shrinking the install partition on Windows, which almost always by default includes the partition that carries all of your files and documents and such – the contents of which used to be known as My Documents.

Therefore, you may need to start moving files off your computer before shrinking partitions and re-organising your hard drive. As we've shown on the partitioning page, you should allow for more space on Windows anyway and you can even use its main partition as your general storage space rather than keeping it completely separate.

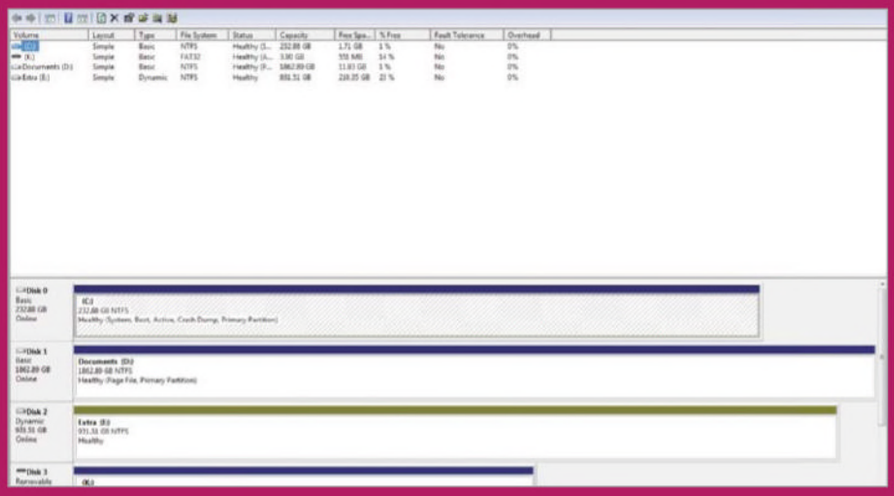
Either way, you need to be careful when shrinking the partition – files usually become fragmented and split themselves up across the disc over the course of many writes and rewrites and moving around. In the days of XP this could become quite ridiculous, however since then Microsoft have implemented auto-defragmentation systems into Windows that generally keep it more tidy on a disc level.

It's not perfect though, and you may feel the need to perform a defrag before shrinking the partition. For this we recommend UltraDefrag (**bit.ly/168iFAr**), which is an open source defragging application. You can make it perform an active defrag that will remove files from the end of the partition – where we will be reclaiming space – and even perform a boot-time defrag so that system files can be moved as well.

Once that is done, you can use GParted to start setting your disc up. Do not move the Windows partition though as it will completely ruin your installation. Instead, keep it where it is at the start and install everything around it.



■ We recommend using UltraDefrag if you want to have a pass at defragmenting



■ Windows has tools for this in Windows Vista and up – you can use those over GParted

Editing GRUB

The boot menu is automatically created by your distros, so it may need some tidying

Modern GRUB, GRUB 2, is very smart. Merely installing it as part of most distros has it look to see what else is on your system and add it to the boot menu. If you've used the GRUB recovery pages in this guide, you'll know that these can be updated at any time to include new distros.

There is a lot more you can do with GRUB though, from simply changing the default selection on the boot menu to customising the naming and placement of operating systems on it. After each update you can save the changes with **sudo update-grub**.

01 Default selection and time-out

On the boot screen the default selection will be the first distro in the list. This will automatically be selected when the timer hits zero. As a quick way of changing the default to better suit your needs, in a terminal use **nano /etc/grub.d/00_header** and then search for the following two lines:

```
GRUB_DEFAULT=0
GRUB_TIMEOUT=5
```

02 Manual order change

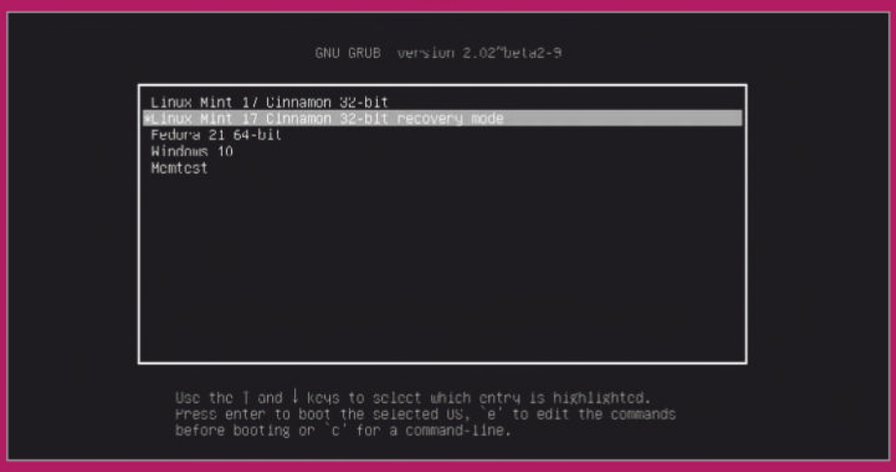
After every **update-grub**, the grub.cfg file is updated, usually located in /boot/grub/grub.cfg. Open it up with nano and scroll down to see the default and time-out changes we made, as well as the individual boot scripts. You can manually move these around in the cfg and save it, but it will be overwritten every time you do an **update-grub**.

03 Quick order change

The order of boot menu placement depends on thev number of the files located in the /etc/grub.d folder. The Linux you're using will have a custom script titled 10_linux, whereas everything else will be discovered using the 30_os-prober list. Changing the number on the two will move them around in the list, for example 09_os-prober will occur before 10_linux.

04 Best order change

The most effective method of changing the order is to create custom scripts for your three distros and order them properly in the grub.d folder. Other distros will use a very similar script to the 10_linux file and you can find a template in 40_custom. Windows is done slightly differently to boot into its chainloader. Once you've got these setup, you may need to do some maintenance on the scripts every few months, but it should keep your GRUB menus in perfect order.



UEFI

When Windows 8 launched there was a lot of furore over UEFI and secure boot. For good reason though, as secure boot would not allow you to install other operating systems alongside Windows. As most PCs and laptops come with Windows as standard, this meant that it would cause major problems for Linux users. Luckily, distros began to adapt and implement software so that even though motherboards still had UEFI and secure boot, they would be able to boot without too much of a problem.

These days there's not a huge problem in installing Linux alongside Windows as most distros have a solution in place, and you can easily deactivate secure boot to get the installation underway. If you do come across any issues though, Google should help you out right away.

"When Windows 8 launched there was a lot of furore over UEFI and secure boot"

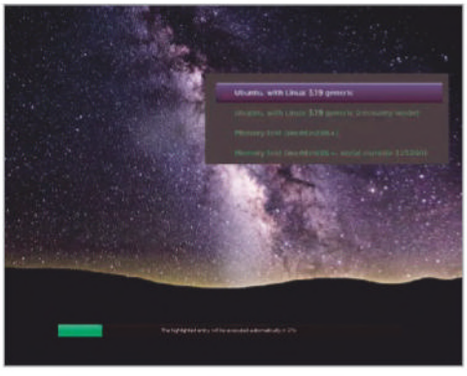
Styling GRUB

If you want to get really fancy, you can always theme GRUB

GRUB isn't, by default, the prettiest thing in the world, so if you're going to be running with your triple boot setup for a while then you might want to give it a little polish.

There are some great guides out there in the wild that walk you through the entire process, including how to set the option titles, the splash image, plus the colours of each element as well as the fonts. Explained in a little more detail just to the right, one great site to check out for this is **dedoimedo.com.**

■ Adding custom backgrounds to GRUB2 is much simpler than you'd think



■ You can also customise the menu options themselves with fonts and colours

Resources

If you need to know more about booting, GRUB and the distros, try these resources

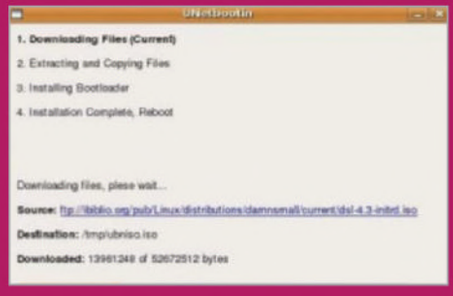
DistroWatch
distrowatch.com

Like the idea of triple booting your system but can't quite figure out exactly what distros to use? DistroWatch compiles one of the most complete list of Linux distributions on the Internet. It keeps tabs on the updates and release cycles of all the major distros, and also has archives of all the update news for each of the listed distros. Every one has its own categories and a little explanation so you can figure out if the distro is what you're looking for before trying out a live disc.

There is a ranking table of distros that seem to be popular on the site, which may help you discover new and excellent operating systems, and an upcoming release schedule so you can plan what distros to get ready to install in advance.

UNetbootin
unetbootin.sourceforge.net

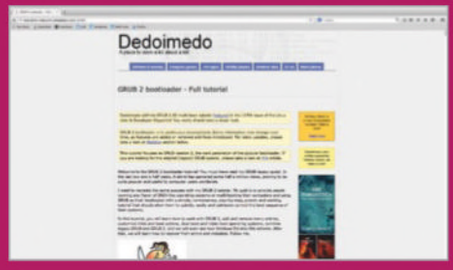
Who has DVDs lying around to burn images to these days? Well... we do! We have a load left that we might need in the future (you never know), but in general we still prefer to use a bootable USB stick to create our live media. Especially when not every computer has a disc drive any more.



UNetbootin is the perfect piece of software to do this and it works across all platforms. It quickly writes ISOs to a bootable USB stick and also has the ability to download a preset selection of distros if you haven't hunted down an ISO yet. It can also add a little bit of re-usable storage to Ubuntu-based distributions. Otherwise, you can simply navigate to the location of an ISO on your system and write the USB from there.

Dedoimedo GRUB 2 tutorials bit.ly/1yoXSk1

A full GRUB 2 reference site can be found at **dedoimedo.com**. It includes a much more in-depth discussion of how GRUB 2 works, how the config file is built and used, the different directories and more. There is also a lot more info on creating your own custom boot scripts for different distributions, a few more recovery tips and ways to customise the look of the boot menu.



It is kept up to date with the latest GRUB 2 changes, so even if you have a problem in the future and need some help, it should remain an excellent resource to go to.

Total Privacy on Linux

Protect your privacy and your personal information from advertisers, doxxers and anyone you may feel threatened by

Your privacy is important. In this modern, always-connected world, finding out who you are and where you are is easier than ever. If it's not advertisers supplying you adverts that are mildly intrusive and slightly creepy, it's intrepid Internet detectives who've decided you have wronged them in some way.

For some, it can be required for an innocent task such as looking for a birthday or Christmas present, while others are driven to mask their personal details by less positive events such as hiding from an abusive spouse or trying to whistleblow without having to fear any repercussions.

Recently there's been a trend of people speaking their mind about controversial topics that have found their personal information displayed on the Internet for all to see. It's no joke – especially when you're threatened publicly and feel like you need to leave your home.

Over the next few pages, we'll cover some of the basics of keeping your information safe, whether you need to do so every now and again or want to make it an ongoing effort.

Long-term privacy

Get yourself private and keep personal details private online for the long haul

The DuckDuckGo search engine does not collect or share personal information

The methods on the previous page are good for keeping your browsing habits at any particular time secret. Privacy is not just about not having any cookies or a known IP address or a lack of Internet history, though – it also applies to your own person. Your address, your phone number and even email address can be precious things that you don't want any random person on the Internet to find out. While you can set up Tor to work permanently on your PC, it won't necessarily keep those details private if you have them anywhere else online.

This may sound fairly obvious, but there are a few ways people can slip up and have their location leaked to the world. One of the major culprits for this is social media, especially services such as Facebook and Twitter.

PRIVACY SETTINGS

Facebook is a big culprit here, with updates causing changes in privacy settings that you'd already turned off. Facebook keeps your email address on file and lets you include it on your profile – you can also add a phone number and full address if you so wish. The best solution here is to not include any of this on your profile at all, but if you need certain people to have access to it, you can easily create lists of friends on Facebook and have it set so only they can see the information.

This applies to other social networks like Google+ as well. Tweak the privacy settings on your pictures, statuses and anything else you don't want the entire world to see.

For a quick way to go through your privacy settings, you can try out PrivacyFix here: **http://privacyfix.com/start**.

LOCATION AWARE

Facebook and Twitter both have location trackers you can use when sending tweets, messages or updating your status. Tweeting publicly that you are at home on Twitter with the location set to on is a good way for someone to track you down.

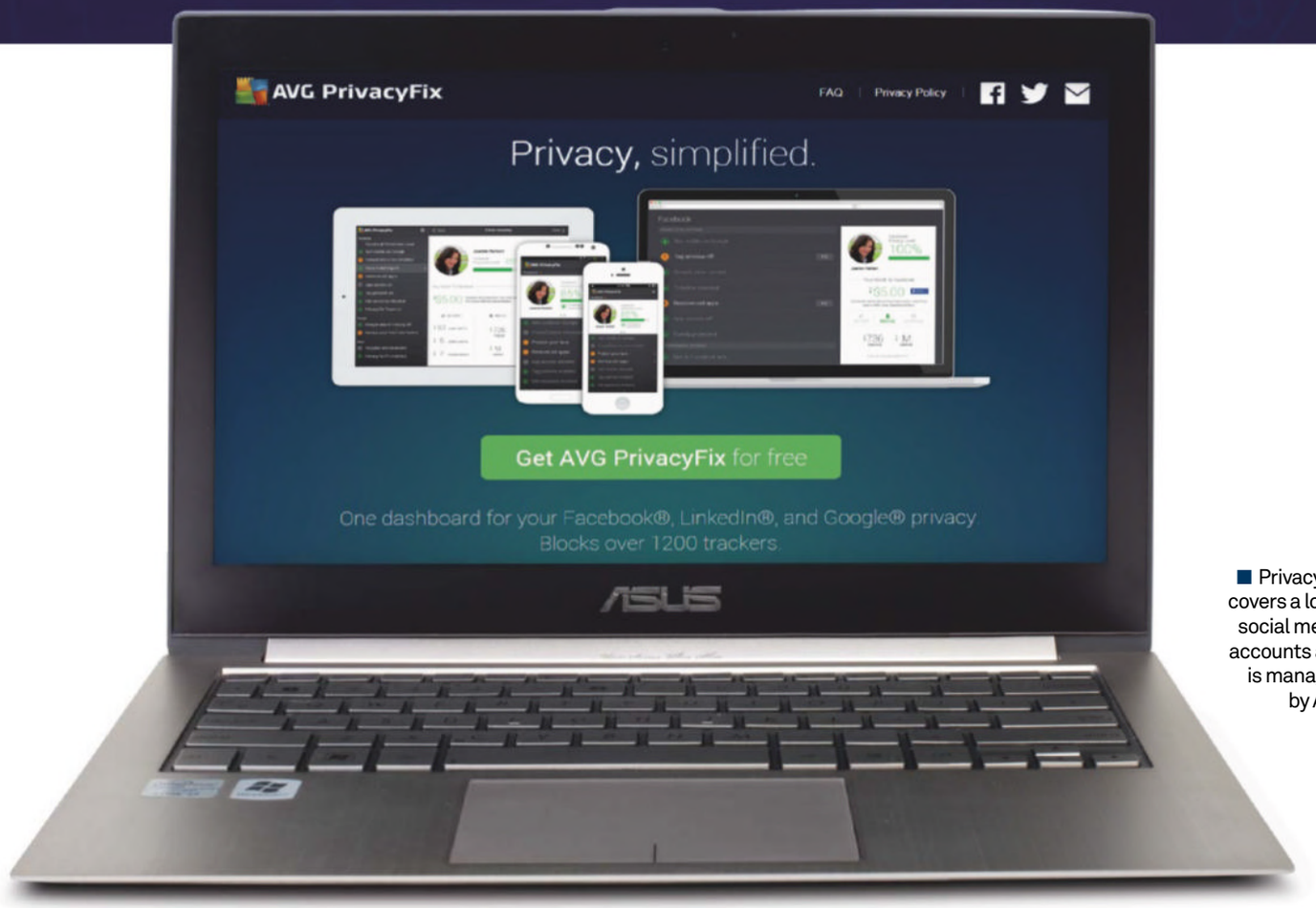
Other things you should be wary about include tweeting pictures of your house or immediate surroundings – with readily available access to Google Street View and satellite imagery, it's easier than ever to figure out where the picture was taken.

PRIVATE WHOIS

Due to laws governing website registration, you're required to supply details for the owner of any web domain. This includes a phone number, address and email contact, all of which will be made public to anyone who knows which websites you've registered. Staying private on the Internet doesn't mean that you don't exist on it, and foregoing a web domain entirely is hardly a good solution.

Real name

Using your real name as little as possible can be good practice. Without your real name, would-be harassers would have very little to go on to start tracking you down. There are some websites that enforce real-name policies (Facebook being a well-known example), but by following our tips on privacy settings this should be less of a concern. Certainly for things like Twitter, you can easily avoid using a real name, along with anywhere else that just requires a username.



■ PrivacyFix covers a lot of social media accounts and is managed by AVG

Most domain registrars now offer a service to make this private, either through themselves or a third party. The information still exists but it is only accessible through this service by people with a warrant, and not the general public. These services do cost money though, however a few dollars or pounds a year can be more than worth the peace of mind it offers.

OLD ACCOUNTS AND APPS

We all have a digital footprint that's years in the making, spanning who knows how many sites that you may have only used once. Luckily some of these may have out-of-date information anyway. Either way, it's good practice to hunt them down and either delete them or replace any sensitive information with false information. Google searching your old usernames and names might help to track down some of the trickier ones to locate, and checking any email archives can help as well.

For older forums or abandoned websites this can be near essential as they can be hacked easier. Speaking of hacking, apps you've approved on Twitter and Facebook and any other social network retain their privileges long after you've stopped using them. Old Twitter clients and Facebook chat apps and quizzes may still be allowed to post on your behalf or have access to your personal information and they're a common target for hackers for this reason.

You should be making periodical checks of your approved apps to make sure any old ones haven't slipped the net. You should also be wary of allowing some of them, especially the Facebook ones, access in the first place.

SEARCH YOURSELF OUT

There are various websites that compile data on people from whatever public information exists. Looking for yourself on these sites can be important as even the smallest bit given somewhere can be correlated back to you. Most of them will allow you to remove your details from their website with little hassle.

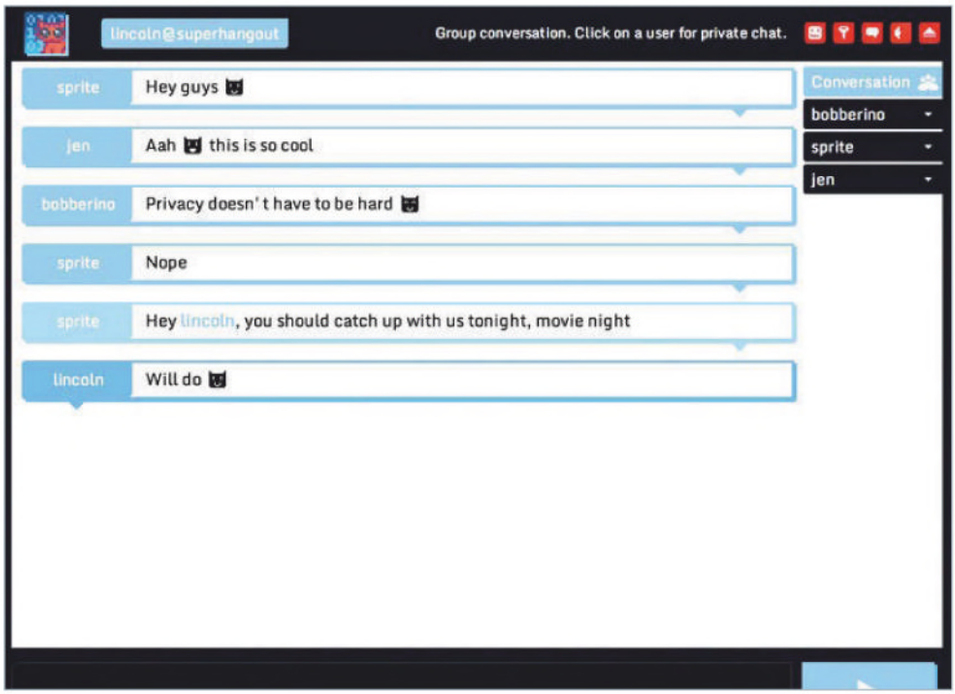
For security tips regarding your privacy, you can also check out Jon Jones' privacy breach survival guide (**bit.ly/1wK4lTo**) for tips on how to secure your accounts, as well as a few more privacy tips.

WHOIS watching?

Making your WHOIS ICANN details private as we've suggested is a good step, but it's unfortunately not perfect. There are sites that keep an archive of WHOIS records, supposedly for cybercrime detection and other related fields, but anyone with $1 and a PayPal account can get a seven-day free trial. They technically shouldn't do this due to certain ToS, but unfortunately there seems to be no way of removing these old records. It's an extra step beyond a quick WHOIS search though that not everyone is willing to do or even knows they can do.

Private apps

Applications you can use to communicate with others that will maintain your privacy



■ Not only does CryptoCat encrypt your messages, it doesn't read any of them either

Worst case scenario

Doxxing is when someone's private information is leaked onto the internet in a malicious attempt to scare that person. It's not a nice thing to have done to you, and it can be hard to get the information removed if it's put up in the right place.

The first thing you need to do is to check what information has been released. Some doxxes end up using old information, meaning you're phone is safe. However, people living at an old address may be mistakenly targeted so you should make sure to inform them. Whether it's old or real information, if it's posted on a third-party website such as Pastebin or Twitter you should immediately report it to be taken down.

It's worth contacting the police to see what they advise you to do in your specific situation, although they may not be able to help much unless you start to get calls harassing you or begin to suspect that someone is watching your house. Remember to stay as safe as possible, and start to hunt down any place that may have resulted in the information being leaked so you can plug it for the future.

Instant messaging

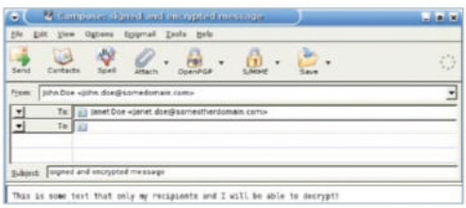
CryptoCat is a secure instant messaging service that works on multiple browsers and smartphones to let you chat with people over an encrypted service. Like a lot of instant messaging clients, you need to make sure the people you want to talk to have the client themselves, but due to the low barrier of entry it's not much of a hassle.

In an article published by ProRepublica (**bit.ly/1wuGegS**), CryptoCat and a few other instant messaging clients scored perfectly on their test. This includes messages being encrypted before transmission, verification of recipients, open source code and security for past conversations or chat logs if something goes wrong. There are a couple other instant messaging clients that scored the top score, so if CryptoCat isn't for you then there are other open source offerings that will do the trick. SilentText, Text Secure and Signal/RedPhone are just three examples which have all had the same score as CryptoCat in the ProRepublica test.

Email

Two of the most popular email clients, Thunderbird and Claws mail, both support PGP encryption on emails. PGP encryption means that only people who are the intended recipients of the email will be able to read them by supplying a special key.

Claws mail supports it by default, but you'll need to install Enigmail for Thunderbird (and Seamonkey) to set up PGP support. This lets you send and receive emails with PGP, and as it's open source it's known to be secure by the community and security vetters. Find out how to use Thunderbird and PGP on page 26.

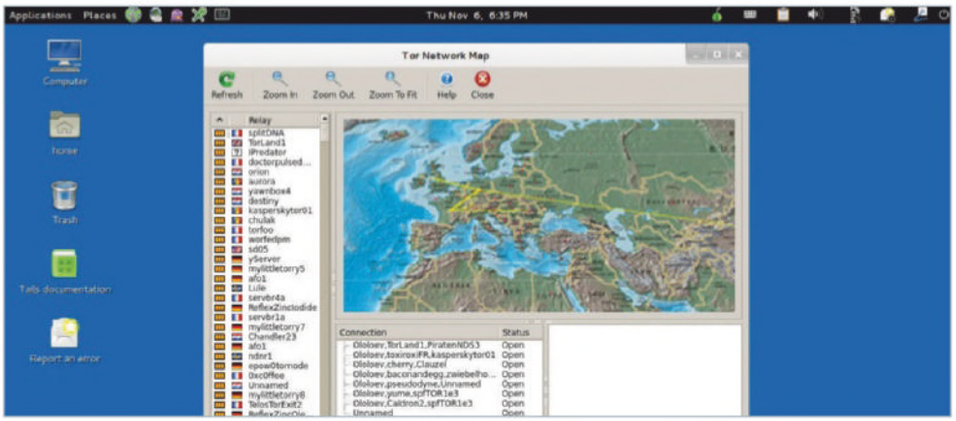


■ Enigmail is a Thunderbird extension

"PGP encryption is used in Tails to make sure emails are private"

Privacy in a hurry

Need to quickly and briefly go anonymous online? Grab the Tails distro for instant privacy



A Sneakers-esque network path visualisation for this particular instance of Tails

Tails is the now-legendary Linux distro that acts to keep your identity, location and activity secret and private. Its entire setup helps you make sure that you can perform whatever task you need to do and leave no trace of your activities on the computer you did them on, and to have any browsing you performed completely obfuscated by bouncing it around the Tor network.

To get it, you first need to download the ISO for the distro from **https://tails.boum.org**.

The ISO can be burned to a disc using something like Brasero, or better yet to a USB stick using UNetbootin. Both of these should be available in your package manager under those names, though if you have trouble finding them then they're easy to find online.

Tails does not install to your system – instead, it works by booting live every time. It lives purely in the RAM so that nothing will get saved to the hard drive. When you perform a shutdown, it will completely rewrite the RAM to erase itself completely from your system.

To boot into it, restart the computer with the CD or USB stick plugged in and look out for the boot menu prompt (usually something like F10). Select the CD drive or USB storage to boot from and it will go straight into Tails. From here you can choose the basic options and you will now be able to browse straight away in total anonymity. However, there are some other customisations as well such as a Windows camouflage mode that looks no different to prying eyes than Windows.

You can write documents, edit images and generally do all the stuff you'd usually want to on a distro, and then send them securely via PGP encrypted emails or other secure online services if need be.

Private browsing

Incognito mode only keeps you anonymous on your own PC

Booting into Tails can be a bit of a hassle if you only need to be fully private every now and then. Tor is readily available to anyone though without going through a specialised distro, and there's a handy Tor browser for your private browsing.

It's based on Firefox and uses the same technology as the Tor button that used to be available for Mozilla's browser. Due to the rapid development of the browser, the Tor team decided instead to create their own spin of the browser. This means they have full control over how it works, guaranteeing that users stay safe and private while using it.

You can get the browser from the Tor website (**http://bit.ly/1jdsLFC**) and it runs directly from the files in Linux without any need for a proper installation. Just make sure that you put the

The Tor Browser has all of Firefox's functions; some are limited for privacy reasons

run command for it in a place that is easily accessible. It's basically the same browser that's in Tails already, so it has a secure search engine and all your traffic is routed through Tor, which makes it untraceable. It's best to not use this as your main browser, due to its limited functionality outside of privacy, although it entirely depends on how you plan to use it.

Troubleshoot & Repair Linux Networks

No network connection on your laptop or problems with your Web hosting? We're here to help

"The Network is the computer," is the famous, prescient quote made by Sun Microsystem's chief scientist and employee number five, John Gage, in 1984. The growth of the web, mobile and cloud computing have borne out that phrase, and a computer without a network connection is just an expensive paperweight.

Fortunately networking is central to Linux, with the Internet, and the Web, having been built on UNIX. Most distros have the built-in tools that will tell you what's going on, or at least start you on your way in investigating your network problem. More sophisticated tools can be found in your distro's repository and, as nearly all of them are command line based, will work just as well on your VPS as your laptop.

Linux puts the power in your hands – you just need to know where to look. Over the next few pages we'll take you through the basics of the GNU/Linux networking stack, and what can go wrong with it (and the rest of the Internet). We'll look at tools and config files to help you and finish with help for using four of the most useful tools: netcat, dig, traceroute and Wireshark.

Network essentials

The first step to troubleshooting your Linux network is to fully understand how it works

Where is the network down? Don't neglect hardware problems – after basic checks it's worth looking for pulled cables or fault lights on your Wi-Fi router – but even following the route your IP packets take, there are lots of places for problems to occur. Some problems are easy to check, while some are more likely than others – let this guide you in the order you tackle your search.

First some background information. You don't need to pass an RHCE (Red Hat Certified Engineer) or LPI (Linux Professional Institute) exam, you just need an appreciation of TCP/IP networking. Feel free to skip through this page lightly, and refer back after reading the more practical parts of the article.

TCP

TCP/IP – Transmission Control Protocol/Internet Protocol – is a set of rules for computers to communicate with each other. TCP sits on top of IP, demanding confirmation for each data packet sent – a lot of overhead compared to UDP (User Datagram Protocol) where no checks are made, but it means there is a lot of useful information available to tools that diagnose TCP/IP problems.

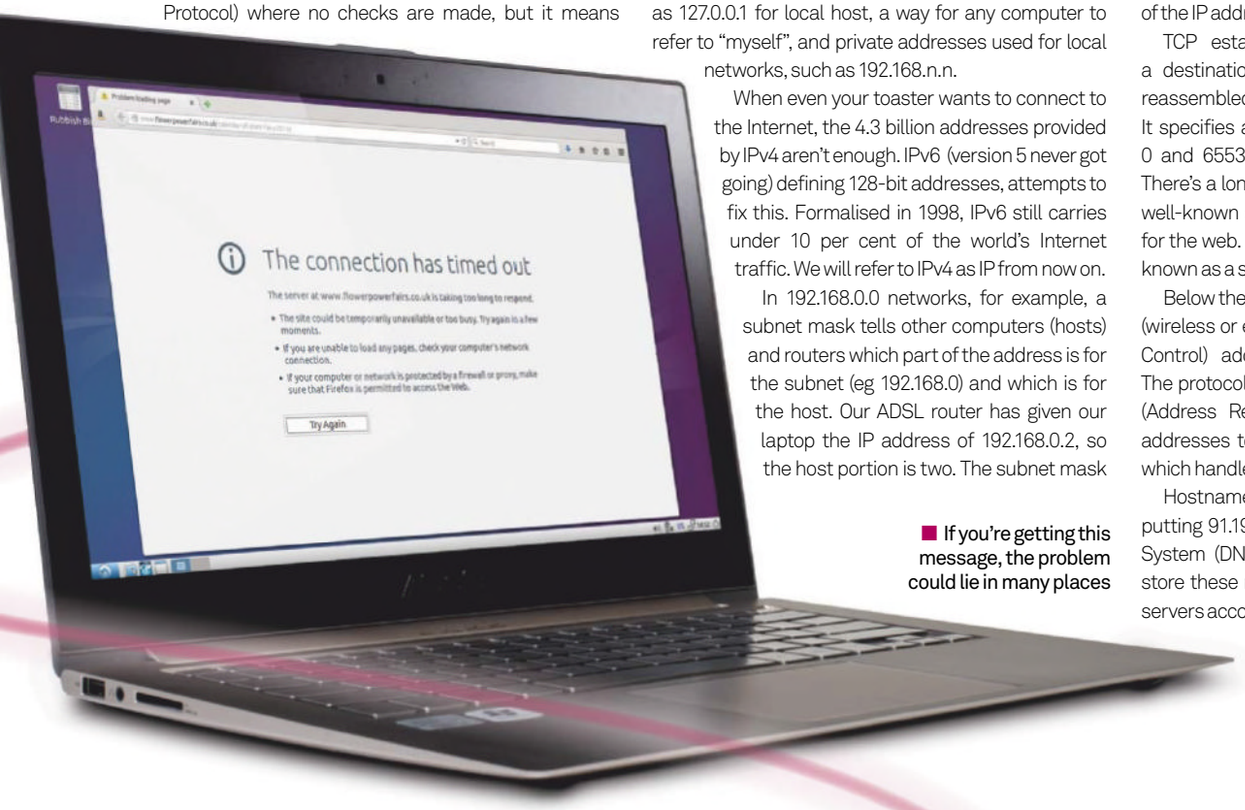
30 years ago, when TCP/IP standards were developed, the computing world was a different place and TCP/IP's independence from the hardware and transmission medium, and open standards and common addressing scheme, have helped give us the networked world we have today.

IP, the Internet layer, defines the datagram – the basic unit of transmission in the Internet, consisting of a header and a block of data. The header contains all the information needed to deliver it – routing from the originating equipment to the destination – in five or six 32-bit words.

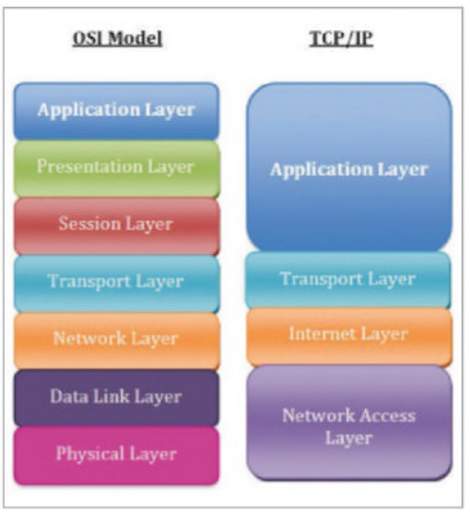
The header contains the destination address for the data. If it's not on the local network, it will be passed to a gateway (or IP router) and continue until it reaches its destination, its journey being determined by routing protocols. The address in IP version 4 (IPv4) is a dotted quad, a 32-bit binary number normally expressed in the form n.n.n.n, where n is anywhere between zero and 255. Certain numbers are reserved, such as 127.0.0.1 for local host, a way for any computer to refer to "myself", and private addresses used for local networks, such as 192.168.n.n.

When even your toaster wants to connect to the Internet, the 4.3 billion addresses provided by IPv4 aren't enough. IPv6 (version 5 never got going) defining 128-bit addresses, attempts to fix this. Formalised in 1998, IPv6 still carries under 10 per cent of the world's Internet traffic. We will refer to IPv4 as IP from now on.

In 192.168.0.0 networks, for example, a subnet mask tells other computers (hosts) and routers which part of the address is for the subnet (eg 192.168.0) and which is for the host. Our ADSL router has given our laptop the IP address of 192.168.0.2, so the host portion is two. The subnet mask is 255.255.255.0, which tells routing devices what parts of the IP address to treat as what.



■ If you're getting this message, the problem could lie in many places



■ Know your onions: compare the OSI seven-layer model with the four-layer TCP/IP model

Credit: CC-SA-3.0 - http://en.wikipedia.org/wiki/File:Application_Layer.png

TCP establishes a virtual connection between a destination and a source, ensuring packets are reassembled in order and re-sending any that get lost. It specifies a port at each end – numbered between 0 and 65535 to indicate the service or application. There's a long list in /etc/services on your machine but well-known ones include 25 for sending mail and 80 for the web. The combination of IP address and port is known as a socket.

Below the level of IP, your physical network hardware (wireless or ethernet card) uses a MAC (Media Access Control) address – six colon-separated numbers. The protocols that deal with this are the ARP Protocol (Address Resolution Protocol), which translates IP addresses to MAC addresses and its reverse, RARP, which handles translation the other way.

Hostnames like **wikipedia.org** are used to save you putting 91.198.174.192 into Firefox. The Domain Name System (DNS) uses DNS servers on the Internet to store these names, and hiccoughs in contacting DNS servers account for many networking problems.

Diagnosing issues

Finding the root problem can be tricky, but there are a number of places you can look first

What's not working – connecting to one website or all of them? If it's just one then it may still be a problem at your end, but if it's everything, let's find out where the problem lies.

First your network connection – most desktop distros ship with NetworkManager to manage connections. From the command line, typing **nm-tool** will report what it knows of your network – look for 'State: connected'. If you don't have nm-tool, use **ifconfig** to see which interfaces are recognised and **ethtool** for connection status information, or use **iwconfig** for wireless connections.

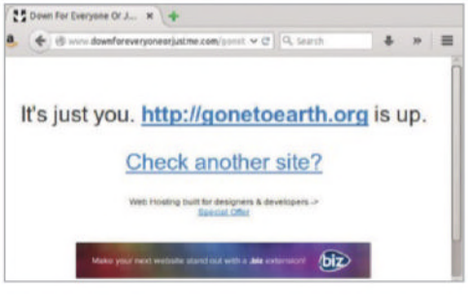
While ethtool will show you're physically connected to the network (Link detected: yes) and iwconfig that you are connected to a wireless router, ifconfig will give you your IP address and netmask, telling you that this much of your networking is successfully configured.

Running **route** will show the routing table, which includes the default gateway to the rest of the Internet. If there's no default gateway shown for addresses outside the local subnet, you will need to fix this. Route can be used to add routes but you need to address the cause of the problem.

Your servers will have fixed IP addresses, which can be edited to correct gateway and other network details. Laptops tend to be configured automatically by a DHCP (Dynamic Host Configuration Protocol) daemon, often running on an ADSL router, where settings can be changed for the problem machine if necessary.

Having corrected settings, a network restart:

```
sudo service networking restart
```

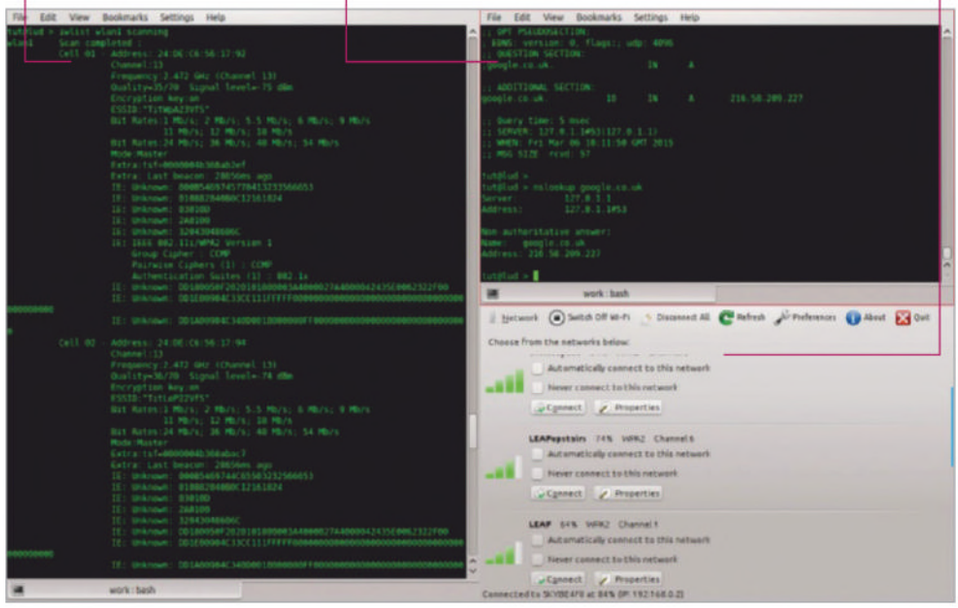


www.downforeveryoneorjustme.com is a very handy diagnostic tool, simple as it sounds

Can't connect to Wi-Fi? The iwlist tool shows you everything your wireless network interface can see

Nslookup gives you the domain's IP address, and where it looked for it. Simple, but effective

A text-based interface and a scriptable version are available for Wicd, but the GUI frontend is fine



...will pick up the revised settings on Debian-based PCs – leave out the gerund (the -ing) for Red Hat boxes. Run route again to check for the appearance of the default gateway.

Ping uses another part of the TCP/IP protocol stack, ICMP, to send an ECHO_REQUEST datagram, and the ICMP ECHO_RESPONSE produced by the host or gateway pinged is used to calculate a time for the trip. Ping tells you if a machine is up, what latency there is in the network and how many packets are lost, all indicative of something unless the server has been set to drop ICMP requests by an overzealous sysadmin, something of negligible security use in most cases.

Use ping to check that you have a route to hosts on the Internet. Start by pinging your gateway:

```
ping 192.168.0.1
```

...then ping a reliable host like 8.8.8.8, one of Google's public-facing DNS servers (the other is 8.8.4.4). We've been using IP addresses and **-n** switches to avoid DNS problems distracting us from other network faults, but now's the time to check DNS functionality. **Nslookup**, less sophisticated than **dig** (part of dig's output can be seen above), but is fine for checking that a domain name resolves to an IP address. If you don't get an answer, have a look in /etc/resolv.conf.

If you've ruled DNS out, try some of the tools overleaf – traceroute to see if you can route all the way there, telnet and friends to see if a particular port is open, dig for more DNS and Wireshark for investigating unresponsive or slow services.

If it is your webserver that's the problem, then **ssh** in and run:

```
netstat -lnp | grep -i apache
```

...(replacing apache with nginx, httpd or whatever is appropriate) to see if your web server is listening to all addresses on port 80. You could **grep** 80 if that's the only port which you're concerned with, but check what else Apache is up and listening on.

Configuration files

Diving into the config files with your favourite text editor is a great way to quickly solve problems

Everything is a file, even connected devices – that's the Linux way. In the Eighties many Unix systems kept binary configurations, but inspired by the Plan9 operating system, Linux put most configuration information in text files. Knowing where they are and what to do with them means your text editor also becomes a powerful tool in checking, fixing and maintaining your Network.

This starts at the hardware level – physical interfaces are found under /dev, and /proc exposes the configuration of installed PCI buses and devices to be read by **lshw** when you call:

```
lshw -C network
```

...to check the logical name entry to use with tools like ethtool and ifconfig.

It's not always simple though. When swapping between Red Hat and Debian/Ubuntu based machines, the ethernet interface on our Ubuntu machine was configured in the file /etc/network/interfaces, while the Fedora 20 laptop's NIC was /etc/sysconfig/network-scripts/ifcfg-em1, sharing a directory with ifcfg-*** files for every wireless hub to which we had ever connected it.

Linux's everything-is-a-file approach also means that if you have issues with hardware, they can often be solved with a text editor. For example, if the kernel isn't loading the module for your NIC then /etc/modules, or a similarly named file on your distro, is the place to add not just modules to load but also alisases to the device's name, if that's what is causing the error:

```
alias eth0 b44
```

DNS again

DNS is accessed by the resolver routines – read the config file /etc/resolv.conf to know where to search. Look at the file on your laptop and you may see something like:

```
As an example your cat /etc/resolv.conf may
# Dynamic resolv.conf(5) file for glibc
resolver(3) generated by resolvconf(8)
# DO NOT EDIT THIS FILE BY HAND -- YOUR
CHANGES WILL BE OVERWRITTEN
nameserver 127.0.1.1
search Home
```

The 127.0.1.1 (rather than 127.0.0.1) is a pointer to a PC running dnsmasq that is a lightweight forwarding DNS server under the control of NetworkManager. In distros without this, dhclient will grab the address of the DNS server from the DHCP server.

It is best to use /etc/resolvconf/resolv.conf.d/base to place an entry like the following:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

...for automatically writing to /etc/resolv.conf. Then running **resolvconf -u** (as root or with sudo) will update resolvconf.

A closer look at /etc/resolv.conf shows it to be a symlink to /run/resolvconf/resolv.conf, which is where dnsmasq writes it. To temporarily remove dnsmasq, try commenting out its entry in /etc/NetworkManager/NetworkManager.conf.

DNS servers are queried in the order they appear in your /etc/resolv.conf file – put the one you want to try out first and/or comment out the remainder by placing a # at the beginning of its line so that the resolvconf ignores it.

Opennicproject.org and **http://freedns.zone** offer DNS with no redirects and no logging, which is essential if you live in a place where what you do online is monitored or restricted.

Rounding off config files by returning to IPv6, it can be removed systemwide by editing /etc/modprobe.d/aliases to add:

```
alias net-pf-10 ipv6 off
alias net-pf-10 off
alias ipv6 off
```

...and rebooting. If you rule it out as a problem, remember to put it back again:

```
alias net-pf-10 ipv6
```

"Smile because it wasn't serious"

Where Am I?

If you are familiar with **whoami**, which tells you which user you're currently logged in as – handy if you su or ssh a lot and risk losing track – you may expect **whereami** to tell you the name of the machine you're logged into. Not so; to do that you type **hostname**, which reads /etc/hostname.

Whereami is a set of useful scripts for detecting which network you've got your laptop plugged into and configuring it accordingly. Particularly handy for those who run lightweight window managers and distros without all the bells and whistles to quickly click on a choice of available Wi-Fi networks, it also lets you tweak known networks with scripts as well as adapt to new connections with minimal intervention.

Running this may help you avoid some connection hassles in the first place, and it's more flexible than wicd.

Is it plugged in?

You may not believe it, but really, these things do happen. It's not the first thing to check – ifconfig and ping will both show that you have a working ethernet connection, or that the Wi-Fi router is up. However, if tests show no connection, that's when you look for loose cables (is the NIC showing a green light?), unplugged routers and any other physical causes.

Don't forget that many laptops have little buttons to switch off the Wi-Fi card (or F-key shortcuts) that can be accidentally pressed. However:

```
rfkill list all
```

...will (usually) show whether or not this is the case. If it's hard-blocked then hit the switch; if soft-blocked then **rfkill unblock all** will usually get your connection up again.

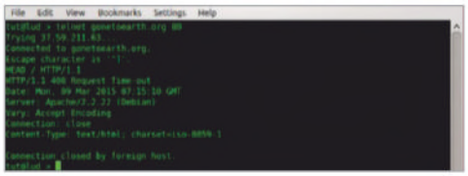
As we said, these things happen, so if it happens to you then just smile because at least it wasn't something more serious that can't be fixed in an instant.

Fix network problems

Using these four apps will help you pin down and fix a number of networking issues

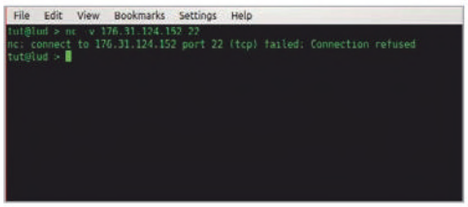
(Telnet to) Netcat

Netcat does everything that the humble telnet does plus much more, but you may find yourself on a box without netcat, so we'll start with an example from old-school telnet.



01 Humble telnet

If you started using computers after the Nineties, when telnet was replaced by SSH in a suddenly far less secure world, you may have dismissed it as a relic from the past. But telnet lives on as a useful diagnostic tool available from any distro, connecting to specific ports to see what's open and working.

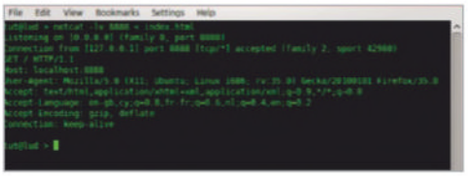


02 Enter netcat

If you can install netcat (nc) then you won't fall back on telnet much, as it combines the simple testing abilities of telnet with abilities to do almost anything with TCP, UDP or Unix-domain sockets: open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports and port scan.

03 Port scan

While it's not good manners to check every port on someone's machine to see what's left open (a portscan), it's useful on your own machines both for security ('that shouldn't be open') and diagnostics ('that should have been open and listening'). Try running something like **nc -vnzu 192.168.0.1 1-65535** to do this.

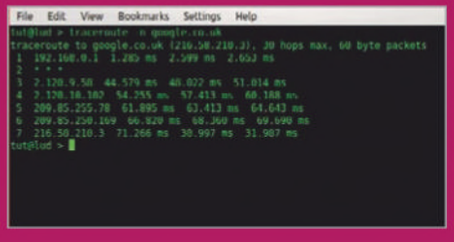


04 Pass the port

One useful nc trick is to quickly set up an impromptu server listening on a particular port, to check there is nothing impeding a connection on that port between you and the server. In the image above, we set nc as a one-off web server and read info on the host that connects to it.

Traceroute

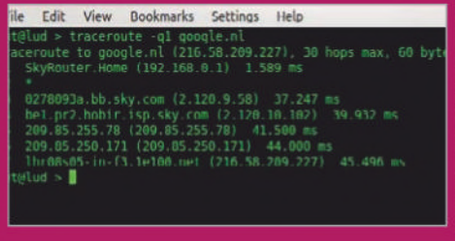
You might not think about how the Internet works while you're using it, but traceroute lifts the lid on where your packets are travelling – showing the time packets take to reach each gateway machine between your machine and the server.



01 Follow the hops

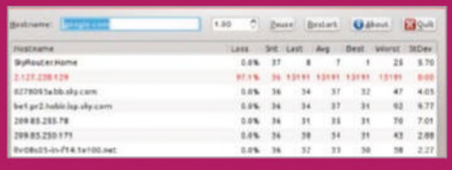
Traceroute tests each hop between you and the destination host. Although not always conclusive, output shows where problems may be occurring. While the screenshot shows the default number of hops and packet size, you can adjust that with:

```
traceroute -m 255 wikipedia.org 70
```



02 Journey times

Those times displayed in ms are the round trip times to each host for three packets sent. Adjust the number of packets with **-q** – for example, **-q1** sends just a single packet. A longer time from the UK could be a channel hop.



03 Mtr

A set of asterixes is an unreachable host but mtr provides a continuous traceroute to help to detect intermittent problems. You may only be able to fix problems found in your own networks, but knowing where the problem lies could help to generate a route around fix.

04 Blocked ICMP

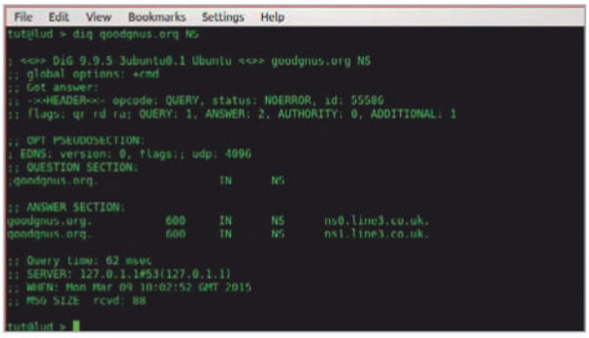
As we mentioned with ping, some systems administrators block ICMP, so standard traceroute won't work. Tcptraceroute provides a traceroute through TCP instead of ICMP.

Dig

We've used the **-n** option a lot in this tutorial, as DNS issues can easily cloud other problems. Once you've cleared up suspected DNS problems on your machine with the resolver, it's time to reach out through the hierarchical world of DNS servers to check everything is as it should be. Nslookup and host perform simple searches, but dig is the most flexible tool available.

01 Address search

Nslookup may be sufficient for resolving an address or checking that you can, but for useful information about DNS servers and their recursive connections across the Internet, fire up dig, whose flexibility means that it repays a little time spent getting to know some of its options.

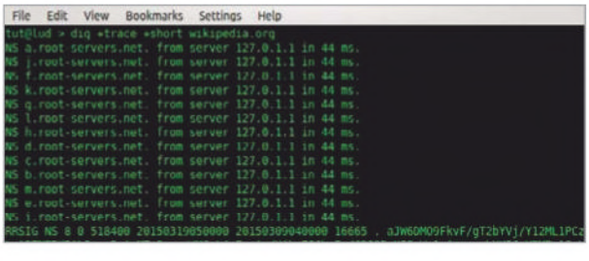


02 Names are served

By default, dig returns A records, but it can be used to check other record types such as MX (mail servers). In the screenshot above we have used NS to find the nameservers for a named domain.

03 Hierarchical

DNS is hierarchical, with the TLD (top level domain), such as .com or .org.uk queried first, then the name part. With searches taking place recursively there's plenty of room for errors – or even malicious attacks. "Dig +trace" shows you the successive hierarchical steps taken by your query.

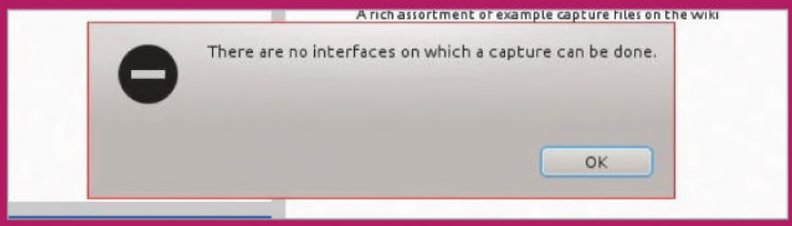


04 +short option

Hierarchical searches output a lot of information that you probably don't need – even from a standard DNS lookup you may only want the IP address. The **+short** option gives you just such an abbreviated output, which is also very useful in scripting searches.

Wireshark

Like tcpdump, Wireshark can dump packets from the network, but it also performs interactive analysis – slightly over the top for minor networking problems, but handy for locating bottlenecks in the system. In most distros Wireshark will be the GUI (Gtk) version, with the console version packaged as tshark. Try them both so you can adapt to whichever is best when trouble strikes.

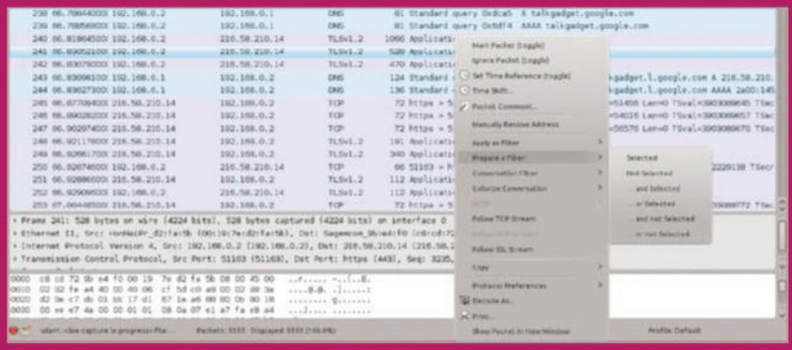


01 Powertool

Despite the baffling number of options available, starting is a simple matter of selecting interfaces from the Capture menu. To get Wireshark to see your interfaces and avoid running it as root user, see Capture Setup/Capture Privileges over at **wiki.wireshark.org.**

02 Portable troubleshooting

As Wireshark is useful for detecting many problems with packet loss or latency, and won't be installed everywhere you go, you can avoid the dance around superuser permissions by carrying it around on a USB live distro.



03 Filter cut

Looking at the raw data is overwhelming and even the choice of filters is large, but you can right-click a suspicious entry and use that as the basis for a filter, or do the same from the filter hierarchy. Simplest case, you're looking for a particular protocol – say DNS, or perhaps something encrypted via TLS – so you just put that in the filter toolbar.

04 Command line shark

On your VPS, or other non-GUI box, tshark is functionally equivalent to Wireshark. It's worth installing after Wireshark and then getting familiar with, so you are prepared if you ever need it.

"Looking at the new data is overwhelming"

BECOME A CERTIFIED SYSADMIN

+ 27 Skills to master for a perfect exam score

We speak to Jim Zemlin and certification experts at The Linux Foundation to find out more about its acclaimed SysAdmin exam – and how you can pass it with flying colours

Linux continues to grow as industry after industry demands faster, more complex technologies to support them. Linux adapts faster than any other operating system because it is open source, built by a global community of thousands of developers and companies. From mobile and embedded devices to cloud computing, supercomputers and consumer electronics, Linux is the fastest-growing platform in the world.

So it should be no surprise that we need more qualified systems administrators and engineers who can support Linux-based systems and enterprises. In fact, our need as an industry is desperate and the Linux Jobs Report underscores this need year after year, reporting that nine out of ten managers are hiring Linux talent every year but that most are having a difficult time finding qualified pros. In the latest Linux Jobs Report (**bit.ly/12yeyfq**), 86 percent of Linux pros said that knowing Linux has given them more career opportunities. 64 percent said they chose to work on Linux because of its pervasiveness in modern-day technology infrastructure.

To address this industry shortfall in the number of qualified Linux professionals that are available to hire, this year The Linux Foundation launched a new accreditation scheme that formalises the Linux SysAdmin and Linux Engineer roles and provides a standard by which potential employees can be measured. Consisting of two qualifications – Linux Foundation Certified SysAdmin (LFCS) and Linux Foundation Certified Engineer (LFCE) – the program centres around an online examination for each that requires candidates to demonstrate their knowledge and skills in a practical manner that is more relevant to the realities of the jobs in question.

The $300 exams are performance-based, testing candidates' proficiency with the command line through a browser-based terminal emulator. Being browser-based, these exams can be taken on any computer and at any time – a key advantage of the accreditation scheme – and are moderated in real time by means of a webcam connection between the candidate and an exam invigilator, with a two-hour time limit for the exam. Furthermore, the exams are distro-agnostic – candidates can choose to sit the exam in Ubuntu, openSUSE or CentOS.

Upon completion of the exam, successful candidates are awarded a digital badge – one of the two shields seen in the System Administrator and Engineer boxouts below – which can be displayed on

System Administrator

What exactly does a Linux Foundation Certified SysAdmin do?

A Linux Foundation Certified System Administrator (LFCS) has the skills to do basic to intermediate system administration from the command line for systems running Linux. Linux Foundation Certified System Administrators are knowledgeable in the operational support of Linux systems and services. They are responsible for first line troubleshooting and analysis, and decide when to escalate issues to engineering teams.

Engineer

How is a Linux Engineer different from a System Administrator?

A Linux Foundation Certified Engineer (LFCE) possesses a wider range and greater depth of skills than the Linux Foundation Certified System Administrator (LFCS). Linux Foundation Certified Engineers are responsible for the design and implementation of system architecture. They provide an escalation path and serve as Subject Matter Experts (SMEs) for the next generation of system administration professionals.

Letter from The Linux Foundation

The demand for Linux talent is real and growing. Linux is now prevalent in highly significant areas such as the cloud, servers, mobile and the Internet of Things, in addition to sometimes less visible but extremely pervasive areas including embedded devices, supercomputers and the automotive industry.

As demonstrated by the annual Linux Jobs Report from The Linux Foundation and Dice.com, 77% of hiring managers had 'finding Linux talent' on their list of priorities in 2014, up from 70% a year earlier, and 46% are beefing up plans for recruiting Linux talent, up from 43% in 2013. However, many organisations in the Linux community have reported difficulty in not only finding qualified candidates, but finding enough professionals looking for these positions to begin with.

This is why in August 2014, after two years of research and consultation, The Linux Foundation launched its first ever certification exams for SysAdmins and Engineers. These exams are distribution-neutral and available to take at any time, from anywhere with a webcam and Internet connection, providing access to many people around the world who could not previously receive a certification simply due to geographic constraints. Coupled with the Foundation's existing training efforts, including the wildly popular and free Introduction to Linux course on edX, the certification program strives to increase the available pool of Linux talent and provide hiring managers with a clear way to determine if a candidate is qualified for a given position.

In just a few months, several thousand people have taken or registered for a Linux Foundation certification exam. This has also prompted many professionals to brush up on their SysAdmin skills by registering for the Foundation's LFS220, a Linux System Administration course, which is now bundled with a SysAdmin certification exam at the end. And to help enable even more professionals to take advantage of the opportunity to become certified, in January 2015 The Linux Foundation will launch a self-paced preparatory course for the SysAdmin certification exam, offering the same content at a lower price and more convenient format.

There is still more work to be done, but training new Linux professionals and certifying them to demonstrate their talent and abilities to employers is one step in meeting the needs of an expanding and diversifying community. Judging by the growing interest and involvement in Linux, the future looks bright indeed.

– Clyde Seepersad
General Manager of Training & Certification

Job hunting and CV writing tips

Training is just the first step – the next one is to find your first job

Once you've got the training, how can you start looking for a job? There are two main parts to this: putting together a good CV and putting yourself out there on the job market. The latter can be done in multiple ways – applying to computing work job sites, setting up a LinkedIn account or sending out CVs to prospective employers.

Look for specialist tech recruiters over general job boards, as you're more likely to find good jobs in the IT sector. As people to fill Linux jobs are in high supply, try and find a site that lets you submit a CV so recruiters can find out about you without getting lost in any mailing system while trying to contact you via third party means. A LinkedIn can also be essential, as many recruiters will search it for prospective employees with the right skills.

As for a CV, keep it short and to the point. One page is ideal: include your essential contact details, computing education and any jobs that are relevant to SysAdmin positions.

■ Jim Zemlin believes that access to learning is more important than ever

■ The Intro to Linux MOOC is free thanks to the edX/Foundation partnership

CVs, portfolios, personal websites, LinkedIn profiles and anywhere else that candidates are advertising their qualification as a Linux professional. With the full strength of The Linux Foundation's support behind them, these small badges will become weighty markers of aptitude in the coming months and years.

As we write this at the tail-end of 2014, the certifications are continuing to make waves as more and more Linux users decide to take the step and sit the exam. The next Linux Jobs Report is due in February 2015 and it will be interesting to see how much of an impact this year's focused recruitment drive from The Linux Foundation has had on the story that it usually tells of our industry's search for talent. In the meantime, however, you can hear more from the Foundation itself about its certification exams and the Linux job markets.

"Based on the four-year trend of the Linux Jobs Report data and the conversations we're having with hundreds of companies all over the world, we expect demand to increase," says Jim Zemlin, executive director of The Linux Foundation, when asked about the predicted demand for Linux SysAdmins in the near future. "Linux is poised to drive technology innovation across industries for decades to come."

Industry demand for talent is one thing, but we're also curious as to whether a pre-existing demand for Linux SysAdmin and Linux Engineer qualifications played a part in the launch of the accreditation program. We asked Zemlin how many people make use of the Foundation's training resources and whether the LFCS exam is something that people had been requesting: "Many members of The Linux Foundation – both individuals and organisations – have told us they would like to see a neutral, highly-regarded certification developed to make it easier to identify Linux talent," he explains. "Additionally, we have seen training course enrolments trending positively, as best demonstrated by the nearly 300,000 people who have registered for our free Intro to Linux course on edX."

Previously a $2,400 course, the renowned 'Introduction to Linux' MOOC (massive open online course) launched earlier this year for free – the fruit of a partnership between The Linux Foundation and edX and, in a way, a test bed for the SysAdmin and Engineer accreditations, with its similar 'anytime, anywhere' structure. Students can opt to fully enrol in the course and pursue a paid-for Verified Certificate of Achievement, which costs $250, or to 'audit' the course, essentially working through the material at their own pace but without any resulting qualification. In the relatively short space of time since the free course launched it has already become wildly popular, and a significant majority of its users opt to audit the course for free.

We asked Zemlin whether, given the historically self-taught nature of Linux professionals (whose CVs often pale in comparison to the wealth of experience permeating their anecdotes), a formal qualification is something that employers necessarily look for: "As Linux has grown and become more pervasive among the world's largest and most technically

■ Attending one of the many events like LinuxCon is a great way to start networking

"Demand for Linux SysAdmins and Engineers has skyrocketed"

complex enterprises, the demand for professional Linux SysAdmins and Engineers has skyrocketed," Zemlin replies. "Certification provides employers with a way to know they are working with the most qualified talent. Certainly many Linux pros will continue to be self-taught; the certification allows them to demonstrate just how good they really are and move them to the next level of their career and earning potential.

"The biggest benefit of LFCS certification," Zemlin continues, "is being able to demonstrate to employers that you are among the best Linux talent on the market. And a Linux Foundation certification is a vendor-neutral, deeply technical program that affirms the credibility of this talent." It's a compelling argument – while Red Hat and SUSE, for example, both offer various Linux training programs, examinations and workshops that are very popular among professionals, this training is necessarily entwined with these two vendors, and beyond the elementary edX course there was no recognised vendor-neutral Linux qualification available to those seeking to bolster their skills and CVs before the launch of LFCS and LFCE. The Linux Foundation is the definitive neutral entity when it comes to Linux, and its official qualifications carry the full weight of its sterling reputation.

We asked how much their exams were informed by similar accreditation programs, such as those offered by Red Hat and SUSE, and the answer was unequivocal: "The exams were informed by The Linux Foundation and a committee of 20 industry experts, from more than ten countries across the Americas, Europe and Asia, to identify the critical skills, knowledge and abilities applicable to each certification. The exam items themselves are written by a group of ten or so external experts and are updated on an ongoing basis to match the required competencies." Elaborating further, Zemlin said, "We are in a unique position to help increase the number of skilled Linux professionals to meet growing demand. We don't take this responsibility lightly and have approached the design of our certification program with attention to the highest quality exams and most rigorous review of the material, which has been informed by a global committee of experts."

It's invigorating to hear that the standards are so high, and that the Foundation is working so actively to address the SysAdmin skill gap in the industry – we'd expect no less. But where does this leave the examinees – is the exam only really viable for existing SysAdmins looking to formalise their experience, or can it also be a springboard for younger talent hoping to secure their first

27 SysAdmin skills to master

Here's what you'll be expected to demonstrate in the LFCS exam, plus the issues of Linux User & Devloper in which you can find the relevant tutorials

Local system administration
- » Creating backups
- » Restoring backed-up data
  - • **113** Backup Masterclass
  - • **121** Full system backups with Clonezilla
  - • **146** Back up to the cloud
- » Managing the startup process and related services
  - • **120** Create and manage boot scripts
- » Managing user processes
  - • **081** SystemTap
- » Creating local user groups
- » Managing file permissions
- » Managing fstab entries
- » Managing local users accounts
- » Managing user accounts
- » Managing user account attributes
- » Setting file permissions and ownership
  - • **139** Sysadmin Masterclass (the above seven)

Local security
- » Accessing the root account
- » Using sudo to manage access to the root account

Shell scripting
- » Basic bash shell scripting
  - • **142** Write useful bash scripts, part 1
  - • **143** Write useful bash scripts, part 2
  - • **144** Write useful bash scripts, part 3

Software management
- » Installing software packages
  - • **086** How to compile software

Command line
- » Editing text files on the command line
  - • **085, 086, 087** A bash at the command line
- » Manipulating text files from the command line

Filesystem and storage
- » Archiving and compressing files and directories
- » Assembling partitions as RAID devices
- » Configuring swap partitions
- » File attributes
- » Finding files on the filesystem
- » Formatting filesystems
- » Mounting filesystems automatically at boot time
- » Mounting networked filesystems
- » Partitioning storage devices
- » Troubleshooting filesystem issues
  - • **111** Perfect Dual Boot (all of the above)

■ As well as networking opportunities, events like LinuxCon provide training workshops

■ 2014 was the year of Enterprise Linux, which means more SysAdmin roles to fill

SysAdmin job? "Most of those who have taken an exam already have had experience as a Linux SysAdmin or Engineer," explains Zemlin. "There is no reason that someone cannot study and prepare then pass the exam without experience in a workplace, though; it will just require more preparation on their part. This is one of the reasons The Linux Foundation is expanding training course options."

According to the Foundation, several thousand people have already enrolled for the LFCS exam, and so far hundreds have successfully completed it; initial pass rates have been around 60%. Given all the feedback that has been received to date, we asked what people thought were the strengths of the exam: "The top strength noted by test takers is that the exam is performance-based rather than multiple choice. This demonstrates actual working knowledge of Linux systems. It also means there may be more than one correct way to answer a question. The goal is not to choose the correct pre-formulated answer, but instead to adequately address a challenge.

"Other than that," continues Zemlin, "exam takers have noted the exams are very comprehensive, requiring them to demonstrate detailed knowledge of a variety of tasks. Finally, the ability to take the exam at a convenient time from anywhere with a webcam and Internet connection has enabled many to take an exam who could not become certified previously without travelling far away to a testing centre. The exams are also distribution-flexible, which test takers have acknowledged is very welcome."

Performance-based testing is highly appropriate to the skills being tested, so it's no surprise that this has been identified as a key strength. It is reassuring, however, to hear that the Foundation's decision to run these exams online is paying dividends, and that people are pleased with both the 'anytime, anywhere' accessibility and the choice of distros that can be used to sit the exam; perhaps, in time, people will begin to request other distros such as Fedora and Debian, but the core selection is sound.

What about limitations, then? We asked whether you could really go into your first job as a Linux SysAdmin upon successfully completing the exam, or whether there are any key areas of the syllabus that would need to be followed up afterwards before you could realistically begin working. The answer was clear and confident: "Generally, if you have the skills to pass the LFCS exam, you are qualified to work as a Linux SysAdmin. Depending on the specific role, you may need more hands-on training, but certainly for entry-level positions the exam provides sufficient demonstration of abilities."

If you're curious as to what those skills are, just take a look at '27 SysAdmin skills to master' on page 29 – this details every subject covered by the LFCS exam, and we've turned it into a reading list of **Linux User & Developer** tutorials for you to work through. The Foundation can help you prepare for your exam, too – returning to the mention of "expanding training course options", our next question was whether the Foundation has any plans to run more live sessions, such as workshops, webinars and even one-to-one sessions. Zemlin's answer was intriguing: "We currently offer training for SysAdmins which can help with the exam; however, in early 2015 we will be launching a self-paced, online prep course bundled with an exam, making course prep easier to access for everyone regardless of geographic location." We can see an 'Introduction to Linux'-style prep course working very well indeed.

While the Foundation does not plan – at this time – to differentiate the course into different skill

"In early 2015 we will launch a self-paced, online prep course bundled with an exam"

■ Continuing Education credits mean that Red Hat training can keep you validated

levels, such as the Junior, Advanced and Senior levels identified in one of its infographics (**training.linuxfoundation.org/sysadmin-evolution**), there are rumblings of new certification paths. When asked about the possibility of other qualifications, such as Linux Foundation Certified OpenStack Engineer (for example), Zemlin said: "Our goal is to develop additional programs for certification but no decisions have been made yet. We are very open to receiving suggestions from the community on areas where new programs would be valuable." Much will hinge on the continued success of LFCS and LFCE over the coming year, for sure, but we can be relatively confident that should the Foundation's new qualifications prove themselves to be industry-recognised badges of quality, this methodology will be applied to other areas of the Linux industry and begin to address demands for proven professionals in other specific areas.

So what's the next step? According to Zemlin, The Linux Foundation will continue to strive to increase access for candidates in order to help them gain the necessary knowledge to take and pass an exam. Furthermore, it will continue to update the content of the exams to ensure that they remain relevant. "Additionally," continues Zemlin, "later this year we will be launching Continuing Education credits that will enable professionals to maintain their certification without retaking an exam by participating in accredited courses, sessions and events."

The Linux Foundation website details two ways in which certification holders can renew their qualification: achieving the higher-level LFCE certification, which extends the expiration date of the LFCS certification to match that of the new one, and completing at least 16 hours of Continuing Education. Continuing Education credits are a means by which candidates can continue their education via The Linux Foundation's training resources while simultaneously renewing their existing qualification, without the need to re-sit the same exam. Currently, there are two primary sources of Continuing Education credits: advanced training courses from the Foundation's Developer and Enterprise curriculums (i.e. those with a Foundation course code of 300 or higher, such as LFD320), and approved training from a Linux Foundation Authorised Training Partner or an established Linux training provider such as Red Hat, SUSE, IBM, Oracle or HP. Any combination of approved courses can be followed, and candidates will need to submit an application (available from the Foundation on request) for the Continuing Education credit that provides evidence of this.

With The Linux Foundation ready to roll out the new prep course and a variety of solid Continuing Education paths to follow already in place, the way forward looks clear indeed. If you are planning to embark on a career in Linux as a SysAdmin or Engineer then there has never been a better time to set out than now – the Linux Foundation is actively looking for you and looking *out* for you on the road ahead, not to mention the employers at the end of that road.

Once you've registered and paid for the exam, you can schedule to take it at any time within 12 months of your purchase – so work through our tutorials and make sure you've mastered the skills that will be tested; identify the areas in which you need further guidance and make use of the Foundation's excellent resources to fill that knowledge gap; read the Certification Preparation Guide (**bit.ly/1vYLKJ3**) and familiarise yourself with the exam setup; then commit yourself and schedule that exam. You'll be glad you did so in a year's time – the industry certainly will.

Get prepared for your exam

The Foundation's Certification Preparation Guide has some very useful tips for success:

Your system The exams are overseen live via your webcam, so you'll need to make some system checks. First ensure your webcam and microphone are working. Then check you're running Chrome/Chromium 32+, as the video feed uses the WebRTC extension. Enable third-party cookies (at least for the duration of your exam). Check your bandwidth next: you need to hit 500kb/s down and 256kb/s up, so temporarily disable syncing and streaming apps. Finally, make sure ports 80 and 443 are open.

Your environment Despite the fact you can take this exam anywhere and at any time, it is still an exam. Switch off your mobile and disable instant messaging, notifications and the like on your system. Make sure you have a quiet, distraction-free environment for the next two hours and be sure to hit the bathroom first. Another thing to note is that you are allowed to customise the standard distro in which you sit the exam; provided you comply with exam rules, feel free to install packages and disable processes (but be aware that this will count against your exam's time limit).

Your exam It's crucial to ensure your government-issued photo ID (like a passport) is handy and that the name matches your Linux Foundation ID Profile on **identity.linuxfoundation.org**, as the person overseeing your exam will check this via webcam. The Foundation's prep guide also recommends practicing looking up **man**, **info** and **help** pages for a few minutes before your exam, to "get you into the rhythm" should you need to look something up during the exam. Also, read the guidelines for using the exam terminal to avoid muscle memory accidents; Ctrl+C/V, for example, is not supported. Once you begin the exam, remember that you don't need to complete the Sections or Objectives in order – feel free to leave questions for later, and use the chat box as a notepad to mark where you were.

Your terminal The terminal you'll use runs in your browser. You can view the guidelines at any time during your exam by entering `man lf_exam`. First, root privileges can be obtained with `sudo -i`. During your exam, do not stop or interfere with the 'gateone' process, as this will end the exam, and also do not block port 8080/tcp. As you'll be inside a browser, remember that Ctrl+W closes the tab; if you need that shortcut, use Ctrl+Alt+W instead. Finally, do not use Ctrl+C/V or copy and paste large amounts of text, as this may result in terminal instability. To copy/paste just small amounts (1-2 lines), select text to copy and then middle-click to paste (or click the left and right mouse buttons simultaneously).

Get rock-solid defences on your systems and networks

PRIVACY BASICS

A couple of issues ago we took a good look at privacy and showed you a number of ways to help protect yourself from advertisers, doxxers and other online threats. That was a great start, and by now you should be browsing online privately, using encrypted tools for online communication and starting to clean up your online footprint. If you missed that feature, just grab issue 147 from our online store: **bit.ly/1sCHWgO**

Linux has a well-deserved reputation for being incredibly secure in comparison to operating systems like Windows and OS X. However, that said, you can't simply rest on your laurels and assume that your computer is impervious to attack – especially in the wake of security scare stories over the course of the last few months such as Heartbleed, Shellshock and the Turla malware, as well as the ever-present threat of more direct system and account intrusions.

This month we're going to tackle security on a number of fronts. First up we'll go through good password practice with a fine-tooth comb, picking out everything that you need to know and showing you how to create super-safe passwords. We'll then take a look at client-side security by running through the optimal settings for your machine and suggesting ways for you to ensure everything important is protected. Networks are next – we'll explain how to build firewalls and properly set up and control your ports, then go through the principles of penetration testing. Finally, we'll return to online matters with a look at securing your various accounts, including using two-factor authentication, and then locking down any information that could potentially be used to hack your accounts.

PASSWORD SECURITY

Creating an invincible password is the first step to securing everything

One of the most important steps in keeping anything secure is to create a very strong password that is difficult to crack. While movies will tell you enterprising hackers just need to look around your office to figure out your password ("it's his son's name – easy"), the most common method of password cracking is a brute force attack on the server and the username.

Under a brute force attack, short passwords, unmodified dictionary words and anything on top password lists will succumb very quickly. In terms of length of password versus time to crack it, the hours and days needed to successfully discover a password are always going down thanks to advancements in CPU speed and bandwidth. Using simple alphanumeric passwords are increasingly insecure, even if they're as long as ten characters.

Let's start with a password then and modify it – a non-dictionary word, reasonably long. Plucked out of the air we have:

dwanton

– Seven characters, lower case letters

Time to crack: two seconds

Seven is quite short. If you're using it online, most websites require a minimum of eight letters, a capital letter and a number. This improves the quality of the password, both off- and online. A basic modification would be:

Dwanton1

– Eight characters, alphanumeric mixture of lower and upper case

There are a few websites that will check your password, but make sure to use something similar

Time to crack: 15 hours

Doing a lot better! The password is immediately exponentially more secure, although 15 hours is still not that long. We can do better by adding a symbol to the mix in an easy-to-remember location:

Dw@nton1

– Eight characters, alphanumeric and symbol mixture

Time to crack: 3 days

Another big jump to three days. In theory, most people would give up by now, but as we're dealing with an automated brute forcing, that won't matter. We're at about as secure as we can be with an eight-character password in terms of brute force, and the '1' at the end is a bit basic. By just making it a two digit number we can further increase the time to crack:

Dw@nton12

– Nine characters, alphanumeric and symbol mixture

Time to crack: 275 days

275 days is quite a while, but it's still doable for persistent crackers. Adding a symbol, letter or number to the end of this password will increase its lifespan to 58 years. 58 years is a massively long time for someone to be trying to crack your password without upgrading their hardware and software or forgetting about it. So here's an example of an excellent starter password idea:

Dw@nton12*

We say starter, as while this is an excellent password, you shouldn't be using it on every account that you have. If a list of passwords is leaked due to someone else's insecurity, it doesn't matter how long your password will take to brute-force if they already know what it is. If you hear of a leak, change your password immediately.

> **"Using simple alphanumeric passwords are increasingly insecure"**

PASSWORD DO'S & DON'TS

DO

- Use capitals, numbers and basic symbols
- Reset online ones every six months
- Make sure it's at least nine characters long
- Run something similar to your password through a password checker

DON'T

- Use dictionary words
- Use phrases
- Use personal information
- Use consecutive numbers
- Use numbers such as your two-digit birth year
- Use the same password everywhere
- Write them down
- Make them too long

ONLINE SECURITY

Effectively use your passwords online and employ other security measures

Now we have a basic password, it's time to start implementing it online. Security experts say you should use a different password for every account. Services like LastPass can offer a convenient way of doing this with truly unique passwords per account, but you might not be comfortable with them. Human beings can only remember so many passwords; as you most definitely should not be making a note of these passwords, what we suggest is to modify the password based on what website you're using it on.

For example, let's take Amazon. It has your credit card details so securing the account is extremely important. After the 'Dw@n' of the 'dwanton' base we have three characters to play with, so we could change them for our Amazon password. Here's our working:

Take the middle three letters of the site's name (as Amazon is six letters long we will choose 'maz'), and reverse the letters to 'zam'. Now insert it into our password:

```
Dw@nzam12*
```

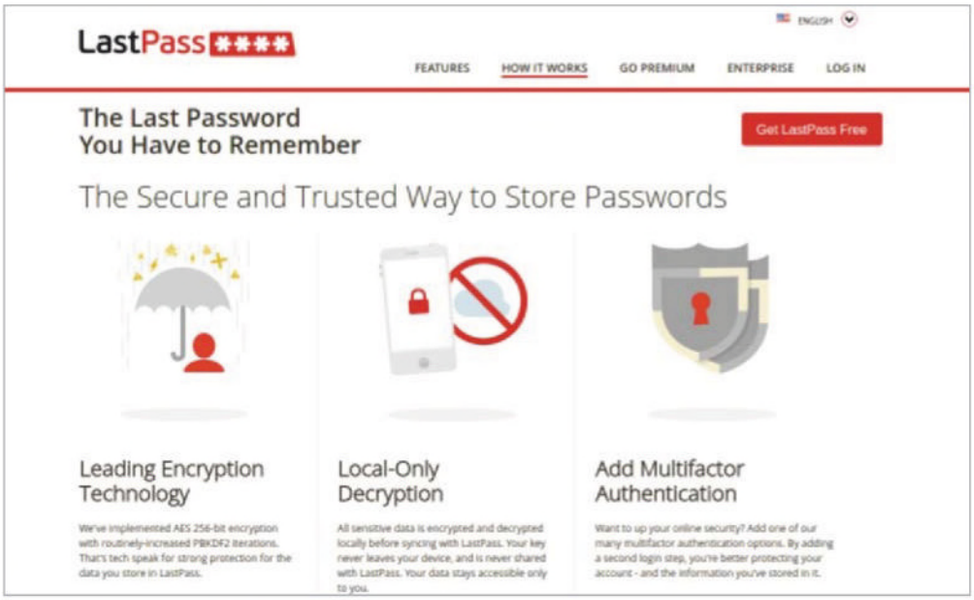
This still has the high level of security, but will be different from, say, eBay (Dw@nabe12*) or Github (Dw@nhti12*), without being immediately obvious to whatever cracking program would then try and use that password on other websites and accounts. A smart enough human might crack the code, but this is only an example of how you can modify your password while still making it memorable to yourself.

Security and Privacy

Last issue we touched upon how to keep your details as private as possible. As well as brute force attacks, crackers can perform confidence and social manipulation tricks with phone support to deceive you, using any information they can gather from social accounts. Some of the privacy-orientated recommendations can help keep angles of attack secret from malicious people.

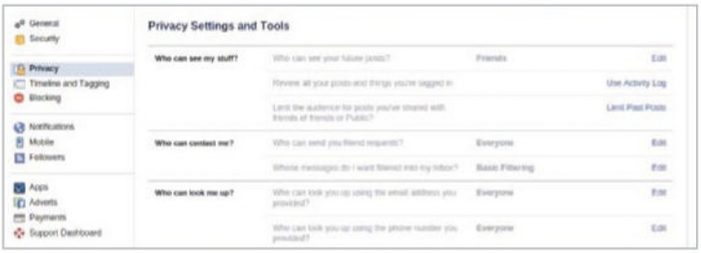
Go through your social media accounts – Twitter and Facebook mainly – and look at your privacy settings. Make sure nothing sensitive is set to private, and even think of removing items that you don't need on your profile, such as phone numbers on Facebook or location on Twitter. Most importantly, keep your main email address completely secret: never share it on Twitter unless via a DM to someone you trust and don't keep it in your profile. For work email, use a different address and link as few accounts as possible to this email.

Lastly, while an extreme step, you can always look at not using your real name on your more public social media accounts. Facebook won't truly allow it for personal accounts, but everywhere else you can it's a good idea to use a pseudonym if you want to be as secure and private as possible.



Above LastPass won our password managers group test a couple of issues back

Below Controlling your visibility on platforms like Facebook is crucial to security



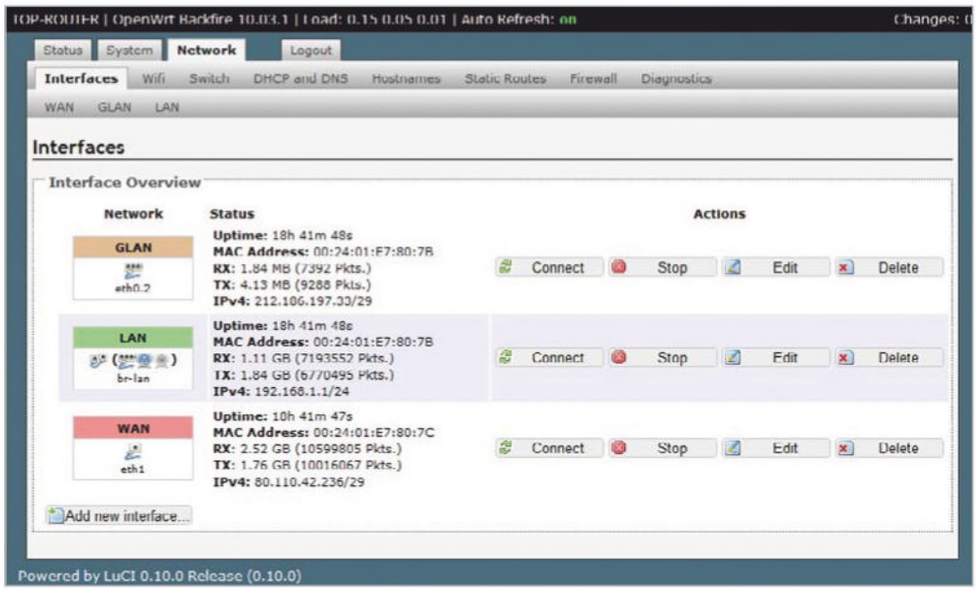
TWO-FACTOR AUTHENTICATION

One of the best security features of recent years is two-factor authentication. Google, Apple, Tumblr and many other services and websites with sensitive information now include two-factor authentication to increase security.

Usually, a mobile number is securely supplied to the company, which then sends you a message including a special, one-use code to work in-conjunction with any system you're trying to log into the account with. Other companies will send it via email or allow you to use a special app similar to banking keypads. Turning this on may be slightly inconvenient to some, but the peace of mind and added security is well worth it.

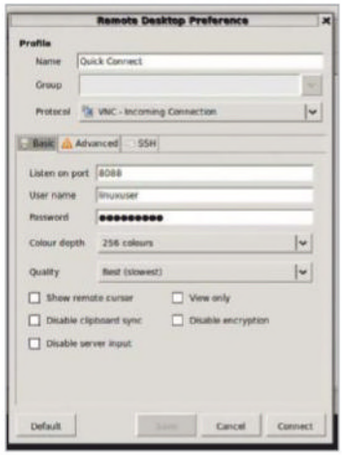
NETWORK SECURITY

Protect your local network from the occurence of external or local intrusions



Below Some tools may be superfluous depending on how you use your system

Left Changing settings in the router can really improve your network's security



Now you've secured your online life, it's a good idea to look at your actual physical network of home or office PCs, laptops and other devices. Having a strong password on your Gmail account is one thing, but if someone can see exactly the kind of packets are going back and forth they can likely figure out what you're looking at. Securing your network is an important step, but it's also fraught with problems you may never know exist, especially when you're trying to balance convenience with security on a system provided by an ISP.

Shared folders, Samba servers and SSH access are very common within networks, allowing you access files and folders remotely. They're also excellent attack vectors by those who can get into your network. Uninstalling or deactivating networking services you don't use is a great way to increase security throughout your network. This is another convenience versus security debate – some networked devices (such as a Raspberry Pi or file server) you may wish to SSH into. That's fine, just make sure that to access them, you require a strong password. Same with the Samba shares for distributing music over your home network and the like. VNC you can probably turn off and on via SSH so you only access it when you definitely need to. There's a lot of thinking about how you interact with devices over the network in terms of network security that can help out.

Network monitoring

What if there's activity over your network that you don't know of and therefore can't immediately fix? One of the best tools in any sysadmin or network security toolbox is Wireshark, or more specifically, the tshark command line implementation.

Wireshark is a network package sniffer and allows you to track all the network traffic going on around your LAN. This can be used to figure out whats going on in your network that you don't know about, stray services and requests and data transfers that either you didn't know about and simply turn off, to finding out some program is transmitting data that it most likely shouldn't be.

Router maintenance

Your router, as the creator of your network, isn't impervious to attacks either. As well as updating any default passwords, you should always make sure to update your router's firmware as regularly as possible. Updates will include security fixes for any vulnerabilities that are present and should improve security across the router's software overall.

If your router allows it, you can also attempt to change the default IP addresses and range. If you're using DHCP, this won't matter to any connected devices, but changing the default network addresses from the common 192.168.x.x structure will stop other types of attacks on the network that specifically target the router.

FIREWALLS

Your router will come with a firewall built in and it will be pretty good. If you have a custom server on your network, either at home or around the office, then you can probably make a better one – one that you have far more control over and that will do a better job of protecting your network overall.

You can do this simply by using 'iptables', a command line tool available for any version of Linux that lets you set up custom rules for IP addresses that can access the server, custom port forwarding and other great changes that make it a lot more useful to certain people.

The Arch and Ubuntu wikis have some great guides: **bit.ly/1yCwG4N** (Arch) and **bit.ly/1gd18ul** (Ubuntu).

OFFLINE SECURITY

Keep your desktop and other offline devices secure from prying fingers

Online security and protection from online attacks are excellent deterrents for a large subsection of people, but your point of access – your PC – should be secure as well unless you want to have people snooping around your computer.

As before, a secure password is essential for your user account. It's a bit harder to brute-force this kind of password, but it's still doable. Weigh up the importance of your files versus your own convenience to come up with a password that suits your needs, but still use the password creation tips to keep it as secure as possible.

The root password shouldn't be the same as a normal user password. Much like we suggested with emails, you should most certainly have a completely separate password for the root account. While logged in as root (su in the terminal) type 'passwd' and it will allow you to update and change your current root password.

Some systems will also allow users to gain root access via 'sudo su', using sudoer privileges to just get into the root. If you're serious about locking down your system then a big priority should be to modify sudoer access on all accounts on your system that can make use of it, especially for that particular use case.

Malware

Linux distros are generally far more secure than other operating systems, but they're not immune to viruses and malware. In terms of security, keyloggers and other snooping software can be a big issue – these will help anyone figure out your passwords, making even the most random 12-character monstrosities pointless when it can just be copied and pasted directly into the password field.

Asides from ClamAV, there's no real anti-malware software for Linux available; the best thing you can do is just stay vigilant. Use a little common sense when on the Internet and check your logs and running services frequently to make sure nothing malicious has installed itself.

Protect your files

Even with a good password, someone can just mount your hard drive or a user with higher privileges can easily read it. Encrypting a volume to specifically keep sensitive data in is a great way to make sure only you can access the files when you need to. Since TrueCrypt has become defunct, and was never really open source in the first place, we highly recommend using EncFS.

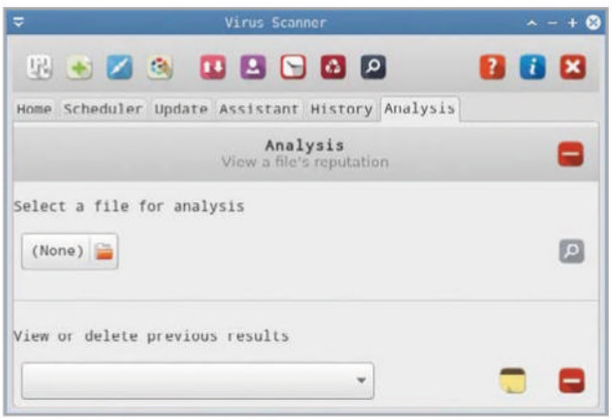
It's available in a few repos as encfs, so installing it is easy. Once that's done, you can then begin setting up an encrypted location on your system. Open the terminal and then type:

```
$ encfs ~/encrypted ~/Private
```

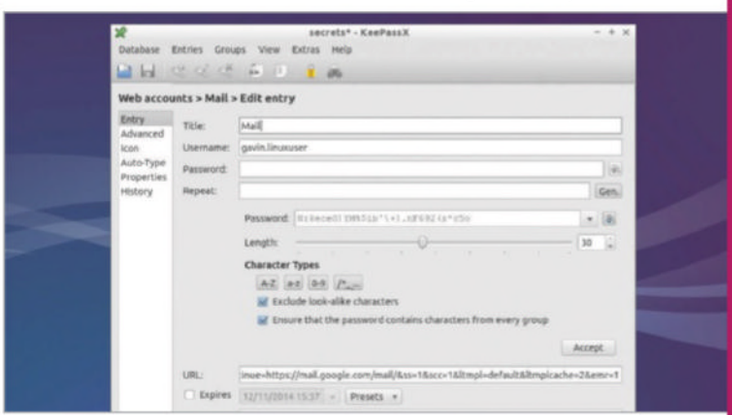
This will create a folder called 'encrypted' in your home directory that contains all the encrypted files, and then another folder called 'private' which is where the files will be accessible once decrypted. Follow through with the little wizard that follows – the preconfigured security mode is very good and good enough for most people.

Now, when Private is mounted and you have entered a password, you'll be able to access the encrypted files straight from the encrypted folder. When it's unmounted, the files will become encrypted once again; just remember to unmount after use.

"The root password shouldn't be the same as a user password"



Above ClamAV is the best you'll find in terms of anti-malware FOSS



KEEPASS X

While LassPass is a password manager for your online accounts, KeePass X is also a password manager for the apps and software on your system. Not only does it keep your systemwide passwords more secure, it also allows you to encrypt information such as URLs, user names, attachments and clipboard items if needed. It has a smart database that you can easily search as long as you have the right password, otherwise it's completely encrypted. It's currently in Alpha release stage for version 2.0, but a lot of Linux users are already integrating it into their system and workflow due to its quality.

SECURITY RESOURCES

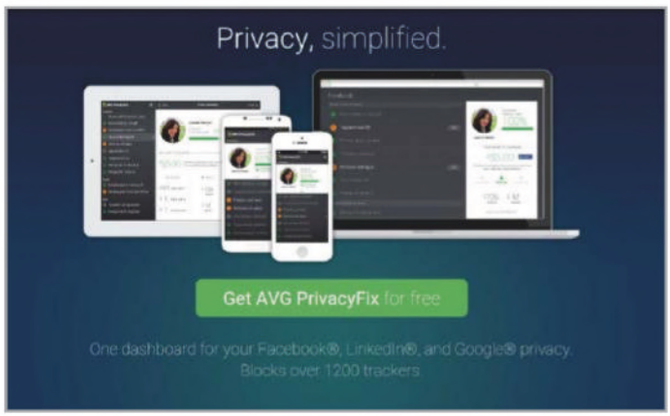
Keep up to date on security concerns and learn more ways to keep your accounts safe

Privacy Fix privacyfix.com

As we mentioned earlier in the article, defending against a brute force attack is only one method in keeping everything secure. Hackers and crackers may also try social manipulation via telephone or email to get your information from banks and account support teams. You can minimise this risk significantly by keep more of your information private.

Privacy Fix from AVG helps you monitor your different social accounts to figure out exactly what people can see and how easily they can see it. It allows you to plug holes in your accounts and tighten up your privacy and security in general.

The applications work across multiple platforms, so you can keep control of these concerns on the go via mobile if something needs to be changed immediately. It covers Twitter, Facebook, Google+, LinkedIn and many other account types.



Above PrivacyFix rounds up your online accounts into a single dashboard

Linux Security linuxsecurity.com

Linux Security is a news aggregate for anything related to security in Linux. Not only does it cover vulnerabilities, bugs in security software and other desktop and server security concerns, it also covers web security and think pieces that will keep you informed on the latest security stories.

Keeping up with relevant issues in the security world can keep you ahead of the game and enable you to lock down anything before a threat becomes viable. It's not absolutely necessary for everyone, but even those slightly interested in keeping secure would do well to keep up with some of the current trends.

There are also some other resources on the site, such as a security glossary for some of the more obscure terms and general security tips that anyone can use.

Above Linux Security is a news aggregate for anything related to Linux

Random.org random.org/passwords

Coming up with a password or password base can be difficult. While we have the example in this issue, we implore you not to use it. However, if you're having trouble coming up with your own base, or want a completely random and secure password for your email accounts, there are lots of websites that will enable you to securely and anonymously generate passwords that you could then slightly modify and use yourself.

Random.org is such a website, where you can create a list of passwords of varying character length that are all very secure. Nothing is stored on their servers and all passwords are sent securely via SSL. The random algorithm uses ambient noise to create the password, which makes it slightly more difficult to decrypt using high-level cryptographic techniques.

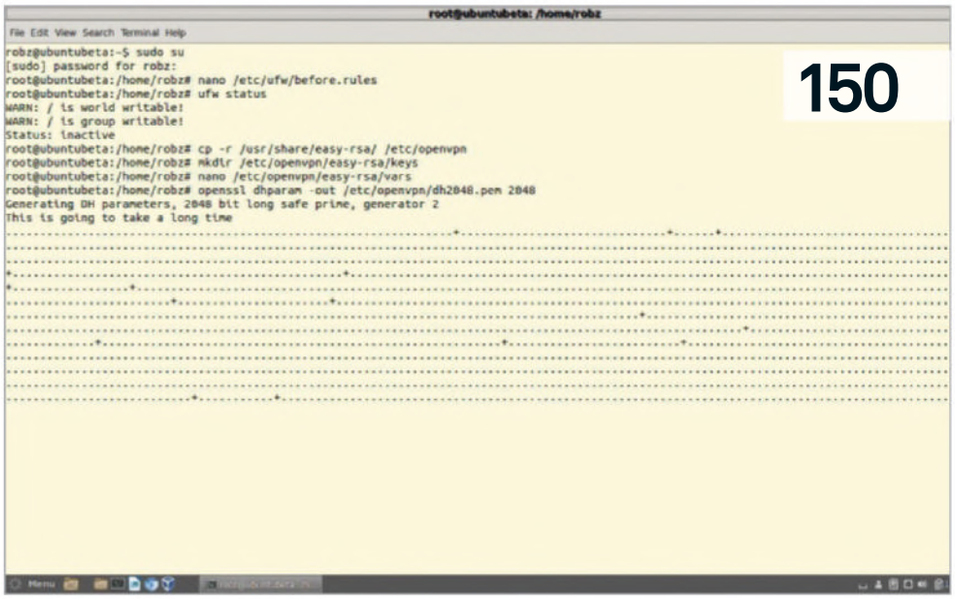
"Coming up with a password or password base can be difficult"

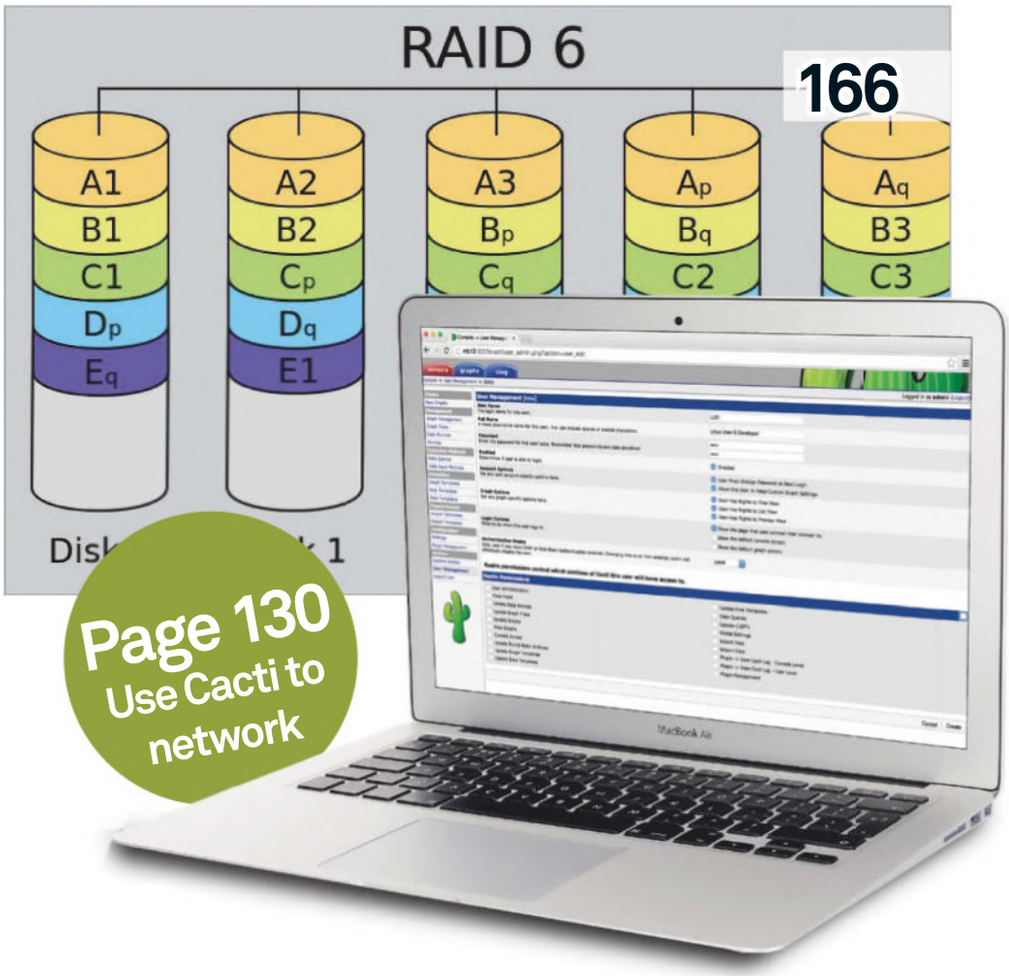
Above **Random.org** comes up with highly secure passwords

Developer guide

Essential tools for coders

"If you want to do something fast, then learn Three.js"

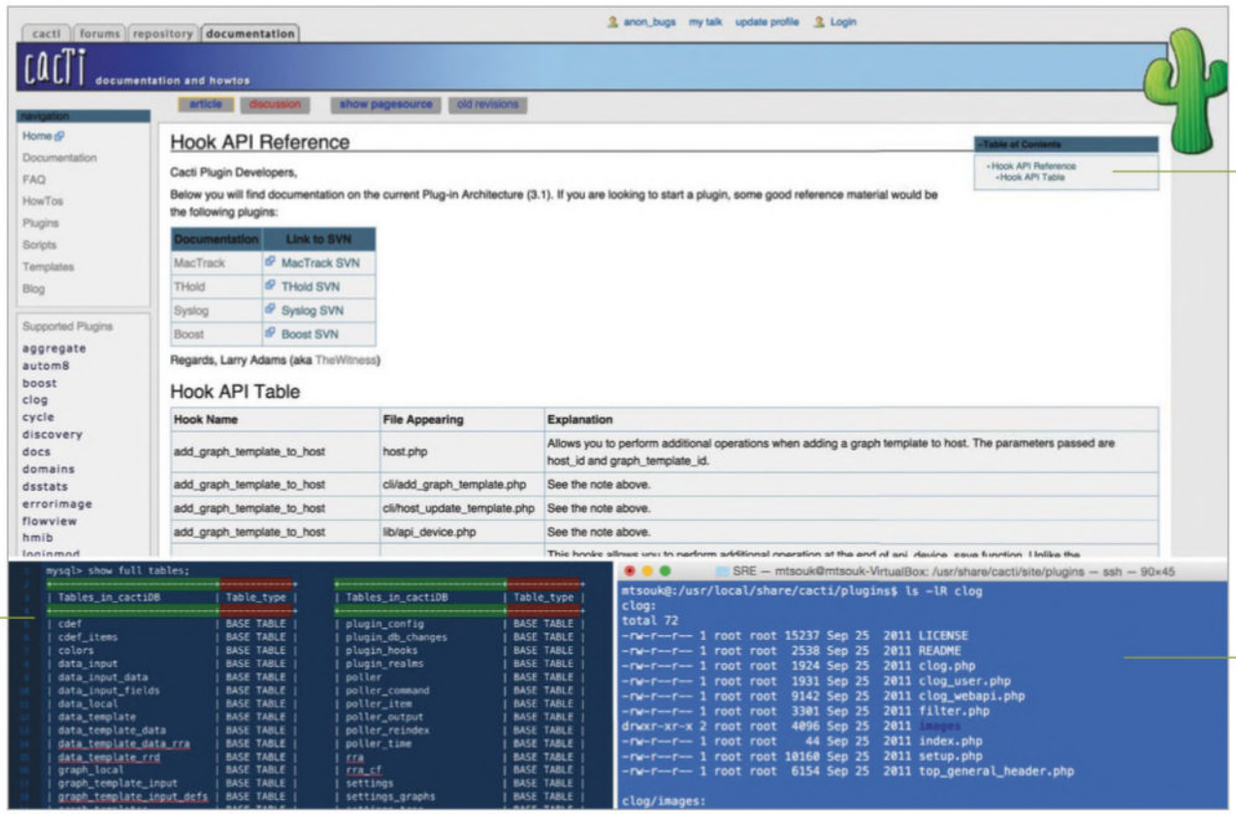





Snowlinux
archlinux
openSUSE
PCLinuxOS
File Edit Torrent View Help
State
Downloading (3)
Active (3)
Queued (4)
Trackers
All (7)
Error (1)
archlinux.org (1)
centos.org (1)
fedoraproject.org (
linuxmint.com (1)
linuxtracker.org (1)
sabayon.org (1)
ubuntu.com (1)
Status
Details
Files
Downloaded:
Uploaded:
Share Ratio:
Next Announce:
Tracker Status:
2m 30s
27/04/12 07:04:09

This is the full list of tables found in the Cacti database. It is worth getting used to them as they hold all Cacti data

Hooks are required when developing Cacti plugins. Consult the Hook API Reference to see which hooks you should use



Each Cacti plugin has a given architecture and structure. This output displays the files of a plugin named 'clog'

Build a Cacti plugin

Advisor

Mihalis Tsoukalos is a UNIX system administrator also proficient in programming, databases and mathematics. He has been using Linux since 1993

Resources

Cacti cacti.net
RRDTool oss.oetiker.ch/rrdtool
Plugins docs.cacti.net/plugins

Develop a Cacti plugin that reads Cacti's database and displays active TCP and UDP connections in a Cacti tab!

In this article you will learn how to develop a Cacti plugin that reads data from the database and then presents it onscreen. You should have a working Cacti installation in order to follow the steps that are described here, be familiar with PHP programming, and have a working knowledge of MySQL so that you can write your own plugins.

The presented example will try to be as generic as possible. The data will be acquired independently using cron, but it will be stored in the same MySQL database as Cacti. You can easily modify the PHP code to read data from an external file or from the Internet.

We would strongly suggest that before you program your own plugins, you study existing plugins – starting with the one presented here – in order to understand the way they work. It is important to remember that all plugins have their own directory and a setup.php file. The setup.php file contains the code that connects the plugin with the Cacti plugin API as well as some other required PHP functions.

As you will see, Cacti allows you to integrate all plugins into its web interface.

getdata.pl — code

02 Register hooks

```
#!/usr/bin/perl -w

use strict;
use warnings;

use DBI;
use DBD::mysql;

my $DBname = "cactiDB";
my $DBuser = "cacti";
my $DBpassword = "cactipass";
my $host = "localhost";

my $connectionInfo = "dbi:mysql:$DBname;$host";
my $connection = DBI->connect($connectionInfo, $DBuser, $DBpassword);

# mysql> desc TCPUDP;
# +---------+----------+------+-----+---------+-------+
# | Field   | Type     | Null | Key | Default | Extra |
# +---------+----------+------+-----+---------+-------+
# | TCP     | int(11)  | YES  |     | NULL    |       |
# | UDP     | int(11)  | YES  |     | NULL    |       |
# | DATE    | datetime | YES  |     | NULL    |       |
# +---------+----------+------+-----+---------+-------+
# 3 rows in set (0.00 sec)
my $query = "insert into TCPUDP (TCP, UDP, DATE) values (?, ?, NOW() ) ";

# prepare your statement for connecting to the database
my $statement = $connection->prepare($query);

my $TCP = `/bin/netstat -nt | tail -n +3 | grep ESTABLISHED | wc -l`;
my $UDP = `/bin/netstat -nu | tail -n +3 | grep ESTABLISHED | wc -l`;

# execute your SQL statement
$statement->execute($TCP, $UDP);

# disconnect from the MySQL database
$connection->disconnect();
```

■ Feeling adventurous? Grab and burn the Windows 10 Technical Preview from **bit.ly/1y8MoE2**

01 About Cacti Plugins

Each Cacti plugin must have its own directory. The central location for all plugins is a directory named 'plugins' inside the Cacti installation. On an Ubuntu distribution, its default path is /usr/share/cacti/site/plugins. As you will learn, it is recommended to move it to a different location.

02 Register hooks

Go to your Cacti installation and select Plugin Management. The list should be empty as no plugins are installed by default. Every plugin needs to register for one or more hooks to use it. You can look into the setup.phpfile of a plugin to find out which hooks it registers. The PHP function that keeps this kind of information is called 'plugin_<PLUGIN_NAME>_install()'. Your plugin must follow a similar practice, so decide which hooks you will have to register. Always register the minimum number of hooks that does the job you want.

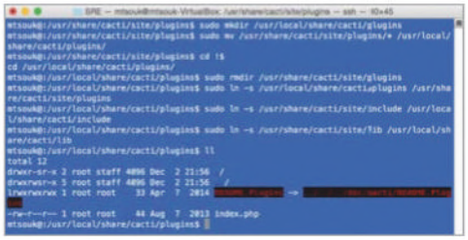
03 Get and store the data

Keep track of the number of active TCP and UDP connections on your local machine, and you'll also need to be able to see the date and the time of each measurement. You're going to need to store the data either in an external file or in a database. The choice is yours, but it would be easier to use a database as Cacti already utilises one. Then decide how you are going to get the data. Using an external script seems to be a generic option.

"The solution is writing your own Cacti plugin to do the job"

04 Explore storage methods

The solution is writing your own Cacti plugin to do the job. This has many advantages, including the fact that you can customise it how you want. For simplicity, the plugin will read data from localhost and display it using a Cacti installation that can also be found on the same machine. A Perl script that is running as a cron job stores the required data in the database. If you find that the method you're using to acquire your data is not suitable for your needs, you can use a different one.



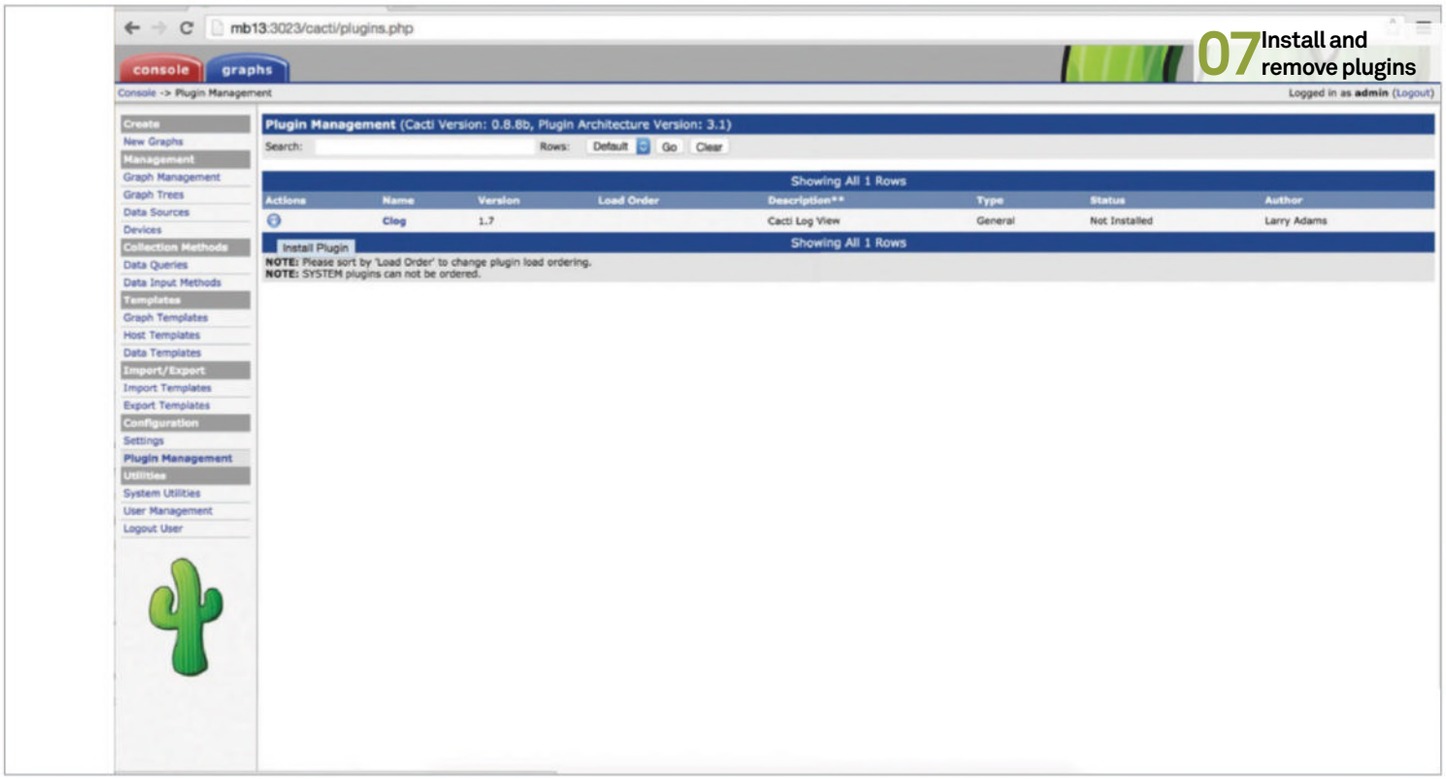
05 Do the required steps

It's considered a good practice to perform some actions before continuing with plugin installation and development. You should create / usr/local/share/cacti/plugins and move all contents from /usr/share/cacti/site/plugins there. Then you should remove /usr/share/cacti/site/plugins, which should be empty after the execution of the mv command. The last step is creating three soft links.

06 Use the Plugin Management menu

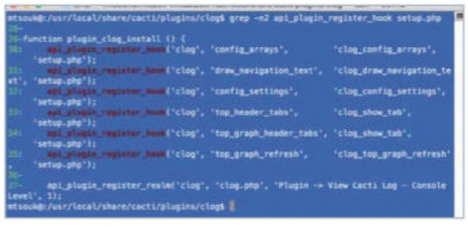
The Plugin Management menu is where all plugin actions take place.

After a successful installation of a plugin, a new tab will appear next to the Console and Graph tabs with the name of the new plugin. You should install and test your plugins one by one, because a broken plugin can break the whole Plugin Management menu. This fact is useful when developing new plugins because it allows you to detect critical errors in code in the early stages of the process.



07 Install or remove an existing plugin

Before installing an existing plugin, you should first download it. Once downloaded, you will be able to install the plugin by putting the files into the plugin directory. It will then automatically appear on the Plugin Management menu. Next, press the Install Plugin option on the Actions column. To use it, you must also press the Enable Plugin button that will appear next. There is also an Uninstall Plugin button that you should use in case you want to remove a plugin at a later point. After disabling, go to the plugins directory and remove the plugin directory manually.



08 Create a new plugin

For a plugin to work properly, two files with given filenames are needed – setup.php and index.php. A plugin may contain more files but without these two, it cannot work. Cacti detects a plugin if it finds a setup.php file inside its directory. There should only be functions inside the setup.php file, therefore don't enter any code that runs automatically. The best way to learn new techniques and improve your own plugins is by reading the code of existing plugins.

09 Develop the plugin

The root directory holds the files and the name of the plugin will be 'connections'. Note that the plugin name must always be in lower case. It is mandatory for the setup.php file to implement particular functions to work. For the connections plugin, the required functions to use are: plugin_connections_install, plugin_connections_uninstall, plugin_connections_version and plugin_connections_check_config. You will also need a function named connections_version(). The best way to find errors or missing functions is to look at the log files of your web server. For example, the connections_version() function can be found in the following error message:

```
[Sat Dec 06 20:19:52.065906 2014] [:error]
[pid 2359] [client 10.0.2.2:62765] PHP
Fatal error: Call to undefined function
connections_version() in
/usr/share/cacti/site/plugins.php on line
303, referer: http://mb13:3023/cacti/
settings.php.
```

10 Finish the connections plugin

After you finish with the basic setup, the plugin is at a working state and has the minimum number of files and functions. At this point, it doesn't return any interesting data but just prints the current time and date. Additionally, it is not properly integrated into Cacti's user interface. If you press the tab it will show the current time and date, but you will be out of the Cacti environment! Don't worry though, as this will be fixed later.

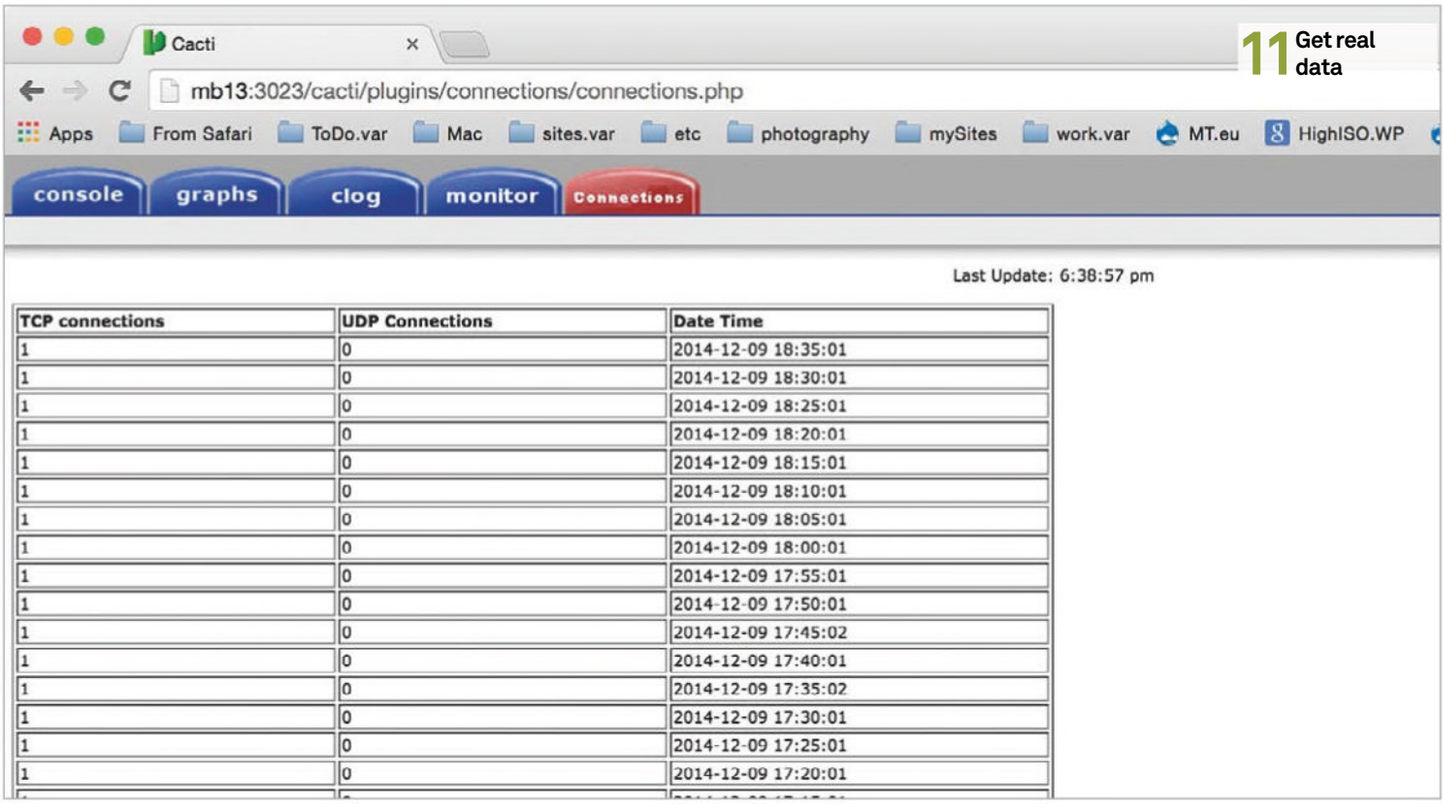
11 Get real data

It's time for the plugin to read a table from the Cacti MySQL database, get the desired data and then print it onscreen using a separate tab. The PHP code will execute a simple SELECT SQL statement to get all data from the TCPUDP table. It will then format the data and present it onscreen. It's important to remember that you can do anything you want with your data. You can display values that are bigger than a given threshold, put colour in the output, and more.

12 Work with the MySQL database

The Cacti database is directly available to your PHP code so you don't have to do anything else in order to use it. The connections plugin reads data from a MySQL database using the following code:

```
$data = db_fetch_assoc("
```



11 **Get real data**

```
SELECT * FROM TCPUDP ORDER BY DATE DESC");
```

All data from the TCPUDP table is stored in the $data variable. With the help of this, it is printed on screen. Being able to work with the Cacti database efficiently is imperative because you can read all the data that Cacti automatically stores there.

"The Admin user is automatically granted permission to use every new plugin"

13 Use important functions

The simple plugin_connections_install() function does not contain too much code, but it is the most important function of the plugin. To understand how a plugin works, you should first look at its install() function. It is populated with api_plugin_register_hook() and api_plugin_register_realm() function calls. The api_plugin_register_hook() function is called as follows:

```
api_plugin_register_hook('connections',
'top_header_tabs', 'connections_show_tab',
'setup.php');
```

The first parameter is the name of the plugin, the second parameter is the name of the hook you want to register, the third parameter is the name of the function you want to call when the hook is triggered and the fourth parameter is the name of the file that contains the preceding function.

14 More details about the plugin code

The general_header.php file integrates your plugin into Cacti, so when you press the Connections tab you will still be inside the environment of Cacti. It is a standard file that you can copy from another plugin. To use it, put the following code inside the connections.php file:

```
include_once('./plugins/connections/
general_header.php');
```

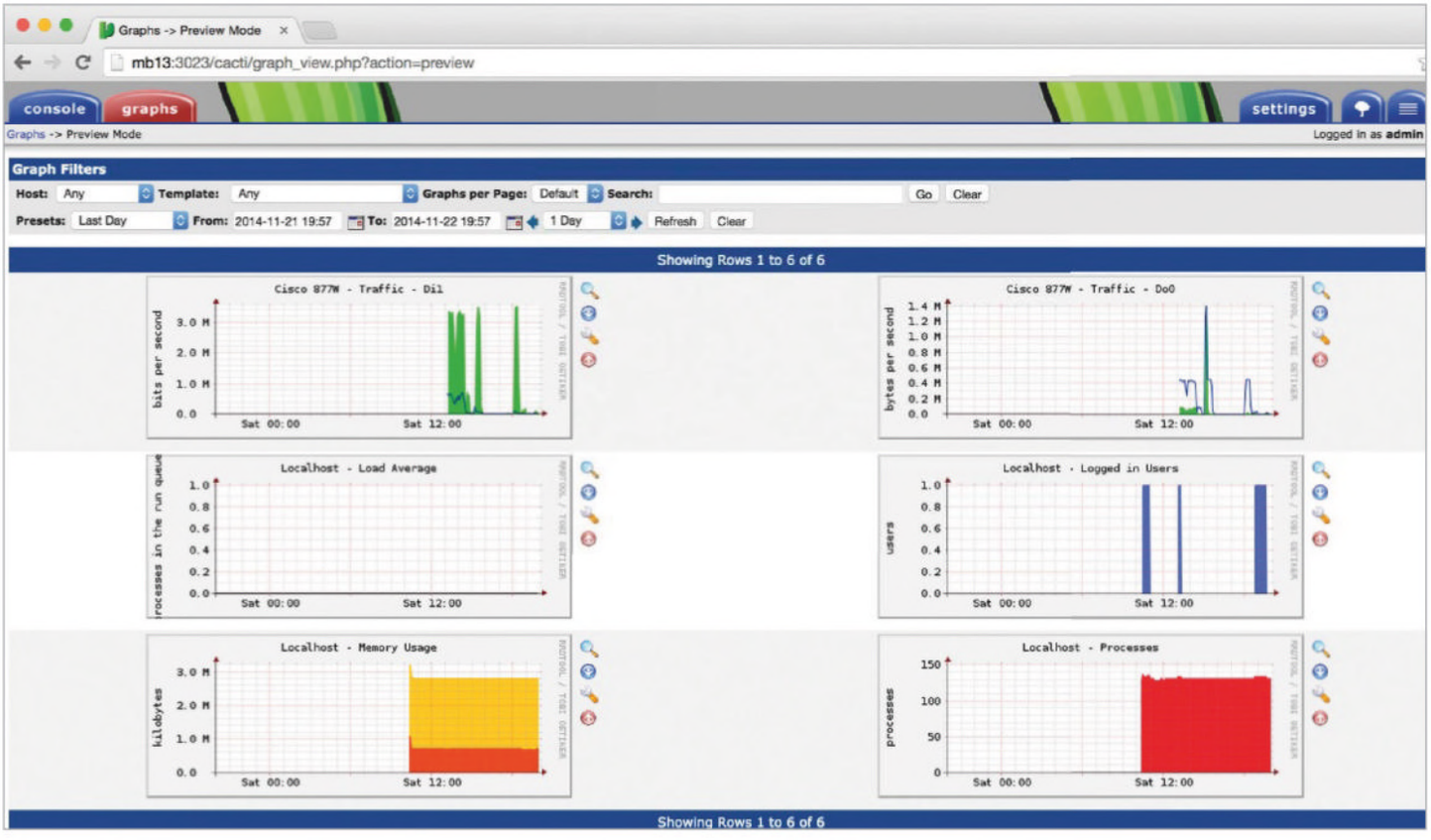
The connections_check_upgrade() function doesn't currently do anything useful. Check the implementation of the upgrade() function from other plugins to find out more.

15 Work with hooks

To display your plugin using a separate tab, implement these two hooks. Both of them usually point to the same PHP function. The top_header_tabs hook enables you to display your own tab along with an image when you're in the console view, whereas the top_graph_header_tabs hook allows you to add tabs to Cacti's user interface. To use the plugin when you aren't in console view, register a realm in your setup.php file as follows:

```
api_plugin_register_realm('connections',
'connections.php', 'TCP/UDP Connections',
1);
```

The Admin user is automatically granted permission to use every new plugin. Even if the administrator is the only one that uses your plugin, you still need to register a realm for it. If you install the plugin without a realm then the tab simply won't appear in Cacti's user interface. Note that if you insert the realm in your PHP code while the plugin is active, you will need to uninstall and reinstall the plugin it for the realm to take effect.



Cacti's Preview view shows all monitoring graphs and is an easy way to get a general overview of what you monitor

Monitor network traffic with Cacti

Advisor

Mihalis Tsoukalos is a UNIX system administrator also proficient in programming, databases and mathematics. He has been using Linux since 1993

Resources

Cacti: http://cacti.net

RRDTool: http://oss.oetiker.ch/rrdtool

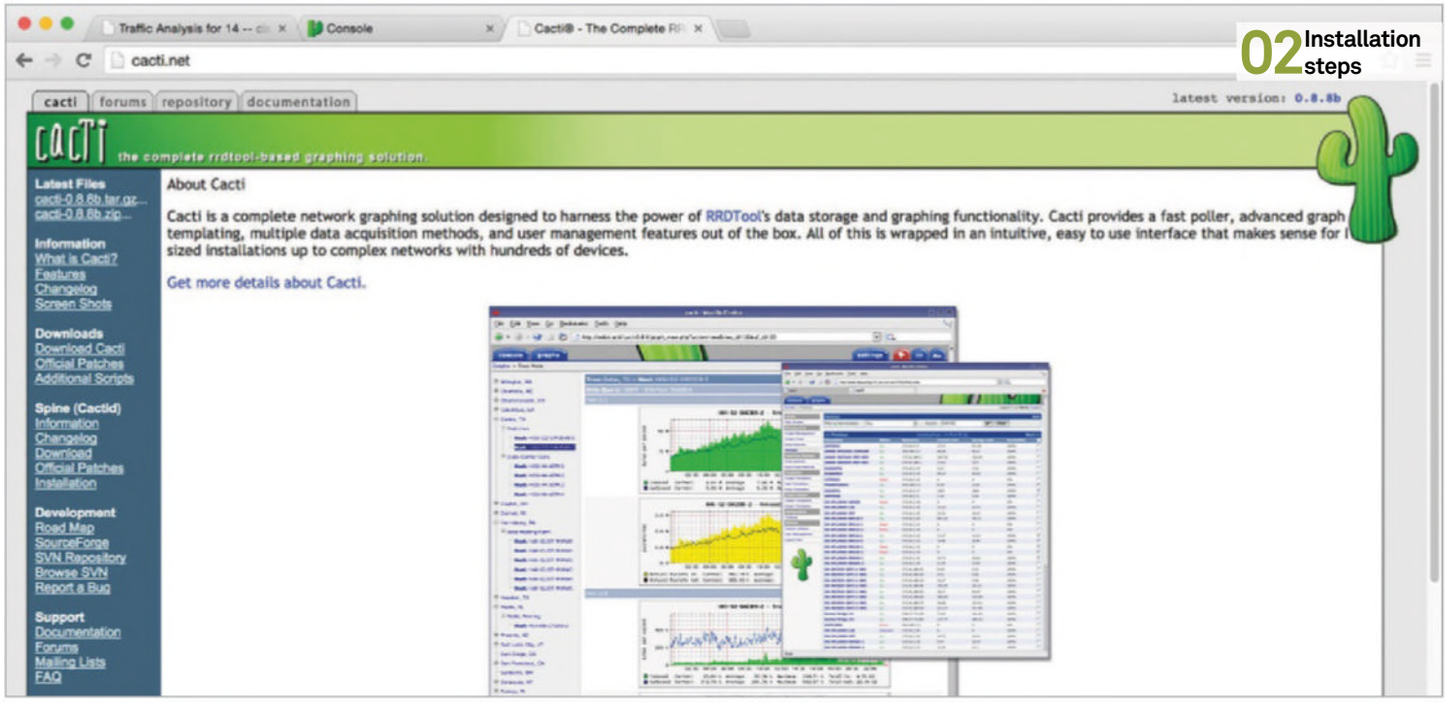
Cisco MIBs: http://bit.ly/1vTeQYA

SNMP RFCs: http://bit.ly/1yv7QUe

Learn how to install and configure Cacti in order to watch the traffic of a Cisco ADSL router using SNMP

Cacti is an open source network graphing application that uses RRDTool – a data logging and graphing system for time series data. A router, by default, connects two different networks, and therefore it should have at least two distinct network interfaces. This article will use a Cisco 877W ADSL router that uses three interfaces: one for the ADSL connection, one RJ45 Ethernet port and a Wi-Fi connection. What you want to monitor is the ADSL connection. Don't worry – do not think that you will need to use the ADSL interface to get the desired data; SNMP can ask any one of the three interfaces and get the same ADSL-related monitoring data!

Our previous tutorial about MRTG in issue 145 of LU&D used the ADSL interface whereas this tutorial will use the IP of the Wi-Fi interface. Using a different IP address or interface makes no difference. Cacti has many more capabilities including support for plugins that enable developers to generate additional Cacti features without dealing with Cacti's source code. A forthcoming article will show you how to develop a Cacti plugin, but for now we'll focus on installation and configuration.



01 Get Cacti

On an Ubuntu system you can get Cacti by running the following command:

```
# apt-get install cacti
```

This command will automatically install RRDTool as well as other required packages. Cacti installs its files at /usr/share/cacti/.

As Cacti is actually a group of PHP scripts and a database working together and creating a monitoring site, multiple devices can be easily monitored using a centralised site.

02 Installation steps

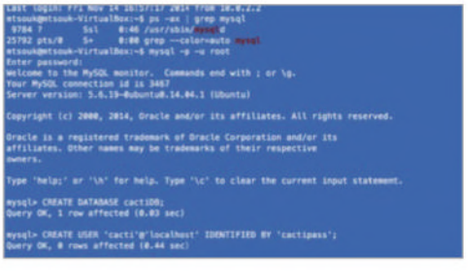
Installing Cacti is far more difficult than MRTG because Cacti uses a database to save its data instead of plain text files. Cacti uses PHP so your Apache configuration should also support PHP. So, you should have MySQL up and running as well as Apache with PHP support before continuing with the installation of Cacti.

The power that Cacti offers does come at a price that you will only have to pay the first time you install it.

03 Pre-installation actions

You should have root privileges in order to set up Cacti on your Linux machine. You should also know the SNMP-related information of the Cisco router. If the ADSL router is not properly configured to support SNMP, you will also need to have administrative privileges on it and set up SNMP yourself.

The Linux machine must also run MySQL. Although it is not necessary, it is very convenient to have a separate MySQL database to store all Cacti-related data.



04 MySQL Setup

Cacti needs a database in order to work and store its data. The default option is the very popular MySQL database. For the purposes of this article, the name of the MySQL user will be 'cacti' and the password will be 'cactipass'. It is good to use a separate database to store all Cacti related data; it will be called 'cactiDB'.

You should manually edit the /etc/cacti/debian.conf file and put in the correct database data. This file replaces the include/config.php file that is usually found in Cacti source files.

05 Apache Setup

The Cacti site will be at the /cacti/ URL as defined in the (default) /etc/apache2/conf-available/cacti.conf file. You will also need to install and turn on PHP support. First install the libapache2-mod-php5 package, then enable the php5 Apache module; on an Ubuntu system, this module is automatically enabled after installation. Otherwise, you will have to either run a2enmod or enable the module manually by editing the Apache config files and restarting.

06 Start Cacti installation

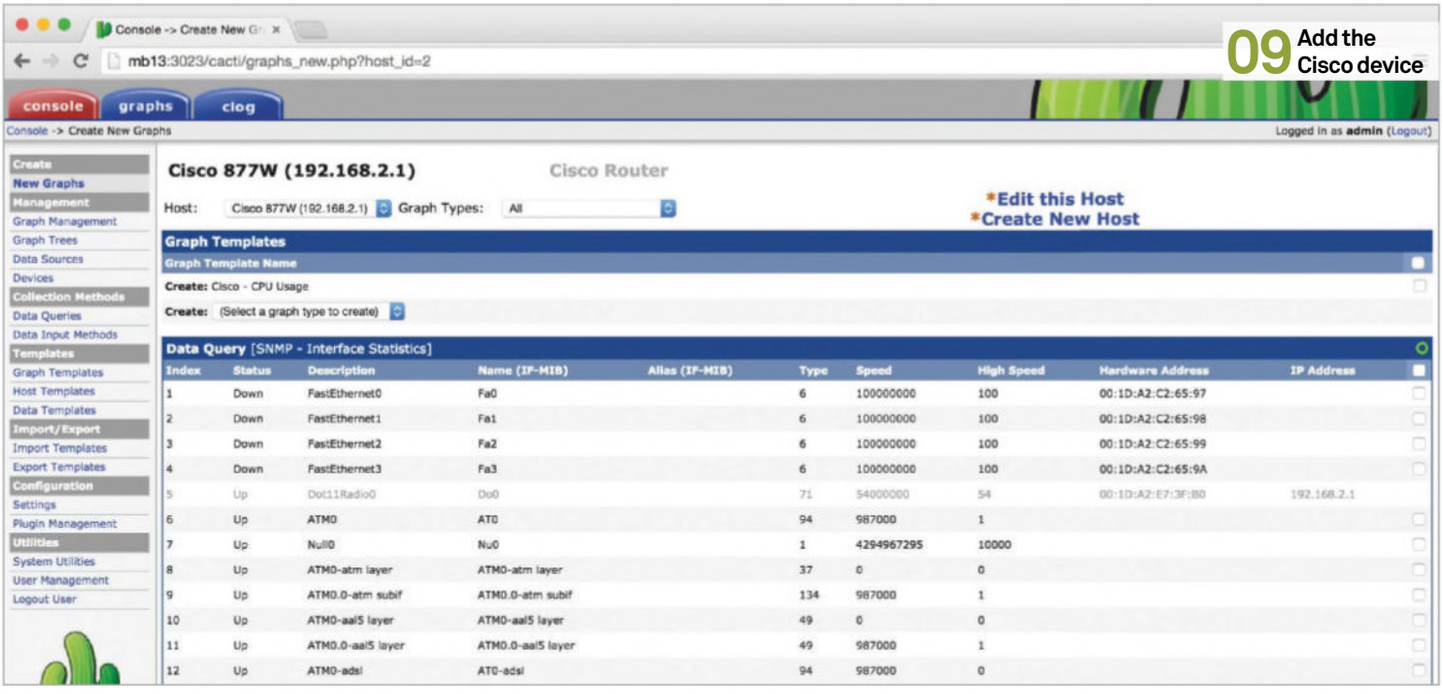
The next step is very important. Before doing anything else you should import the Cacti database data inside the MySQL database. On an Ubuntu Linux System, this can be done with the following command:

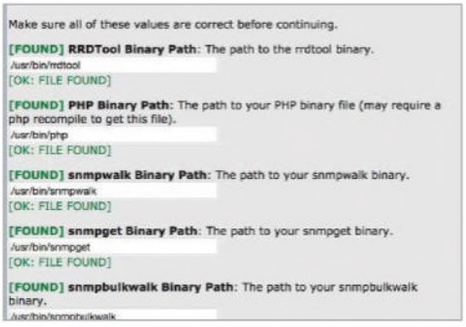
```
$ cat /usr/share/doc/cacti/cacti.sql |
mysql -u cacti -p cactiDB
```

Without this step, the installation will fail and you are going to get no data from Cacti!

In order to start the installation process, you should now point your favourite browser at the /cacti/ URL. We will use the **http://mb13:3023/cacti/** URL but yours will vary.

"Cacti needs a database in order to work"





07 Finish the installation

Two more simple steps will be required that should present little to no difficulty. After finishing all steps, you should log in to Cacti using the admin user. The first thing you will be asked to do is change the default password for the admin user, which is also "admin". From now on you can use the **http://mb13:3023/cacti/index.php** URL to connect to your Cacti installation.

08 Check Cisco configuration

SNMP is a known TCP/IP protocol that is available for most 'clever' devices, including Linux and other UNIX machines, routers, network switches, Windows machines, etc.

Before you start installing Cacti, you should make sure that the device you are going to monitor is properly configured. Cacti will acquire data from the Cisco router using SNMP, so you should check if SNMP is properly working on Cisco.

09 Add the Cisco device

After installing and configuring Cacti, you are ready to add devices and graphs to Cacti. The things that you need to know in advance in order to monitor your Cisco router with Cacti are the name of the SNMP community (LUD) and the name or the IP address of the router (192.268.2.1).

To add the ADSL router, you should first click on Devices on the left. Then select Cisco Router and 'Enabled' on Status, and then click Add. Now you will have to fill in the required information that depends on your configuration. It is important to put the correct data in the Hostname and 'SNMP community' fields. Then, click Create.

If everything is okay, the next screen will display 'Create Graphs for this Host' on the upper-right side of the screen. Click on the 'Create Graphs for this Host' link to go to the next screen.

The next screen after this lists all the available Cisco interfaces for this particular router. What interests us right now is Interface number 14 (Dialer1), which is the ADSL Internet connection interface, so make it active. The desired graph type should be 'In/Out Bits'. Now click the Create button. Other interfaces of interest may be Number 5 (Dot11Radio0), which is Cisco's Wi-Fi interface and Number 13, which is the Ethernet interface. Now select Graph Trees from the left menu and then click Add. The next screen will allow you to select the device you want to monitor. Follow the instructions on-screen, and you are done!

10 The output generated by Cacti

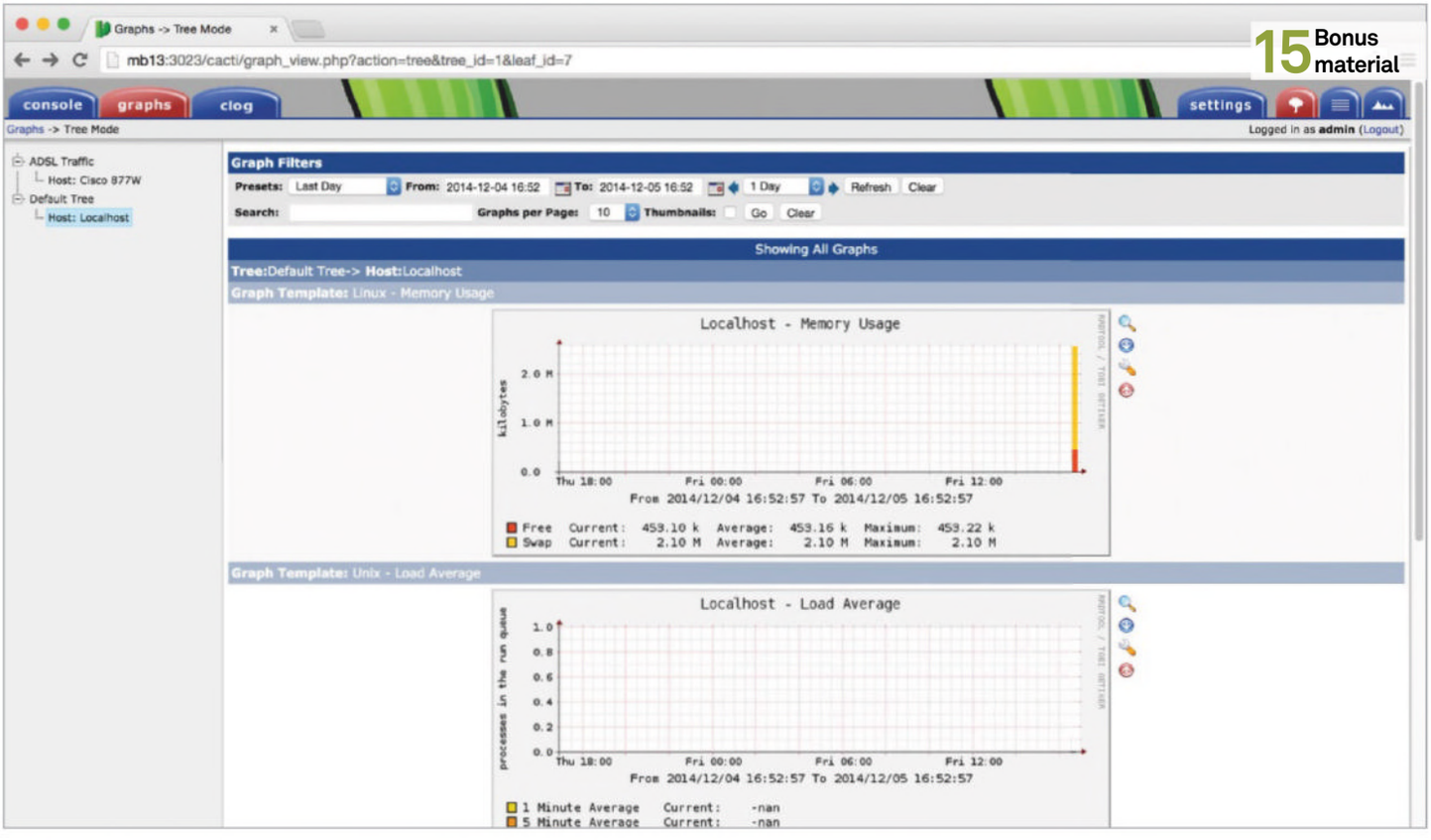
A script that runs as a cron job updates the output of Cacti – this is automatically configured during installation. You can look at the /etc/cron.d/cacti file for more information about the way Cacti is being executed.

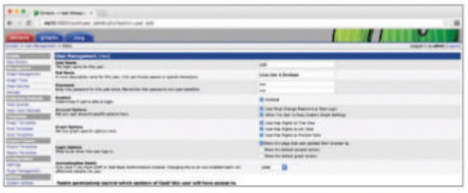
Select the Graphs tab and then, from the Default Tree, select the desired host. You will have to wait a little, until some data is obtained in order for the graphs to be populated.

11 User management

Users in Cacti can be divided into three brief categories: Anonymous, Normal and Administrators. What differentiates these three categories is the way they authenticate and the permissions they have.

You can also add users using the command-line interface of Cacti but using the graphical interface is simpler. You can visit **http://www.cacti.net/downloads/docs/html/scripts.html** to find more information about Cacti command line scripts.





"Although backing up MRTG is a simple copy process, backing up Cacti is more demanding"

12 Create and use a new User

Head across to Console>Utilities>User Management. You can see that two users are created by default: admin and guest. Click Add to create a new Normal user called 'LUD'. Fill in the required information. On the Realm Permissions, turn on the View Graphs checkbox. Now, press Create to create the new user. It is always useful to turn on the 'User must change password at next login' option.

After user creation, edit the new user, go to the Graph Permissions tab and add the graphs that you want the user to be allowed to see.

13 Cacti directories

The /var/lib/cacti/rra directory contains all the RRD files that keep your performance data. The /usr/share/cacti/resource directory holds all the XML files responsible for the data queries of Cacti. The /var/log/cacti directory contains all Cacti log files of Cacti – you should visit its files when there are problems with Cacti. The /usr/share/cacti/cli directory holds all the command line scripts.

14 Back up and restore

Although backing up MRTG is a simple copy process, backing up Cacti is more demanding because its data is stored on a database. You can manually backup the MySQL database using the following command:

```
$ mysqldump -u cacti -pcactipass cactiDB
> cacti.sql
```

The generated cacti.sql plain text file contains SQL commands that can reconstruct a database from scratch (including its data).

Similarly, you can back up all Cacti configuration files using a simple UNIX script; just make sure that you include all files and directories. If you are not that familiar with MySQL, you should ask your database administrator for help, because backup is a critical task.

15 Bonus material

The default Cacti installation automatically gathers and displays monitoring data about the current machine (localhost). You can see it by selecting Graphs>Default Tree>Host:Localhost.

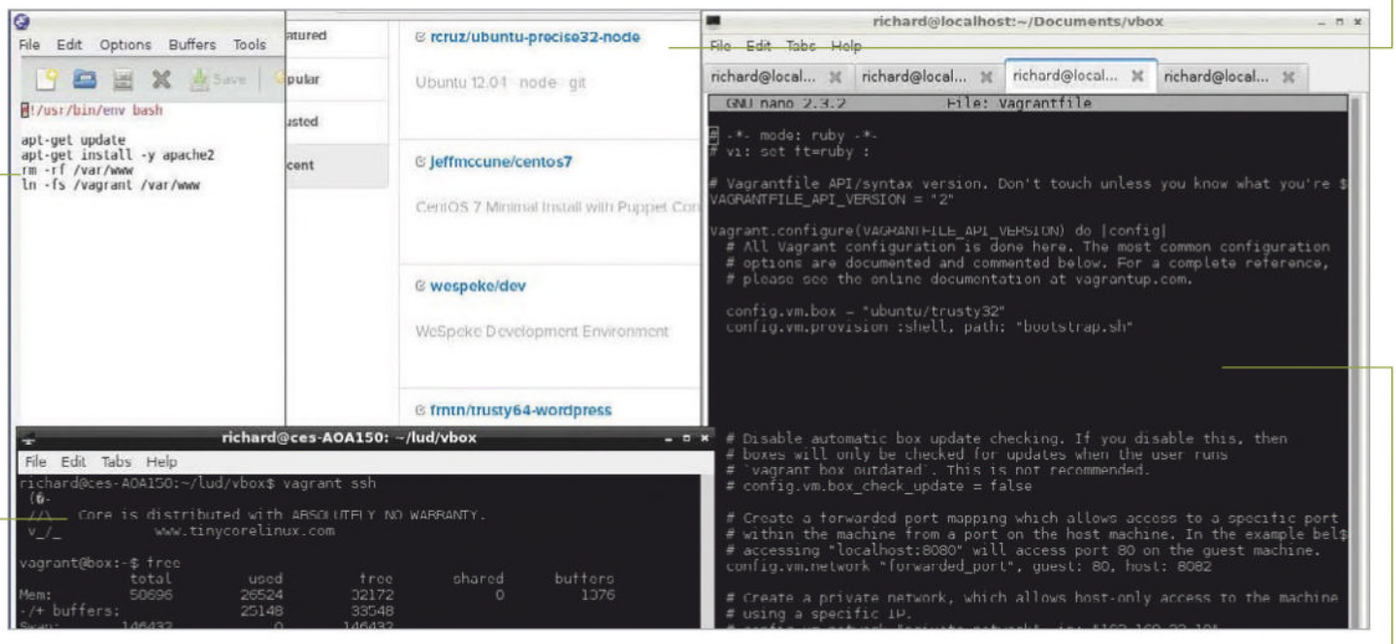
16 Final thoughts

Installing Cacti is not as easy as installing MRTG. Nevertheless, Cacti produces a more professional output, supports plugins and multiple users with different privileges. Cacti also supports templates. They are used for simplifying the creation and administration of graphs.

Depending on your needs, you can choose MRTG or Cacti and be assured that any of them will serve you well.

Vagrant makes it easy to provision boxes with any development and deployment setup you need

Hundreds of base images are ready-made and additional software is ready-configured too



You will be in a working virtual box from scratch after just three vagrant commands (init, up and ssh)

Share your Vagrantfile and all your coworkers will have the same development environment on all platforms

Configure virtual boxes with Puppet and Vagrant – part 1

Make it simpler to develop all kinds of server apps and manage the deployment of new servers by using virtual machines

Advisor

Richard Smedley A Unix jack-of-all-trades, Richard doesn't spend enough time in any language to get truly proficient, but always has a shell open so learnt scripting by osmosis

Resources

Ruby ruby-lang.org
Virtual Box virtualbox.org
Vagrant vagrantup.com/downloads

It may not ever be 'The Year of the Linux Desktop', but free and open source dominates the boxes where web apps live, so how do we develop for them across a heterogeneous environment? Vagrant holds together VirtualBox or any other virtualisation software – it works with Amazon EC2 and VMware, and can work with containers like Docker and OpenVZ. It can also work with various config tools to make an easy-to-manage, portable development environment.

Its greatest advantage is eliminating differences between development and deployment environments, drastically reducing unnecessary errors. As your needs grow more complex, Vagrant's close integration with config tools like Puppet will lift the admin burden from your shoulders.

Share the single config file (Vagrantfile) with your team, with or without Puppet invocation, and everyone will have the same environment on any platform.

Those who are hooked on using the Puppet config tool will need no excuse to throw its configuration management powers at any appropriate problems. We hope we can convince the rest of you that it's worth learning in conjunction with Vagrant, but this month we'll get you going with Vagrant alone. First, let's make sure we're speaking the same language by updating your Ruby installation.

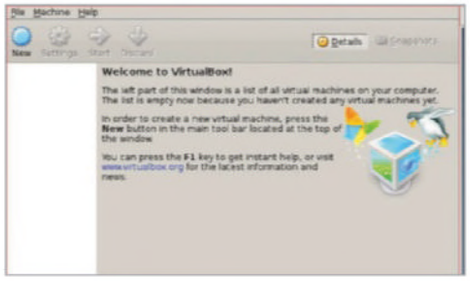
01 Ruby

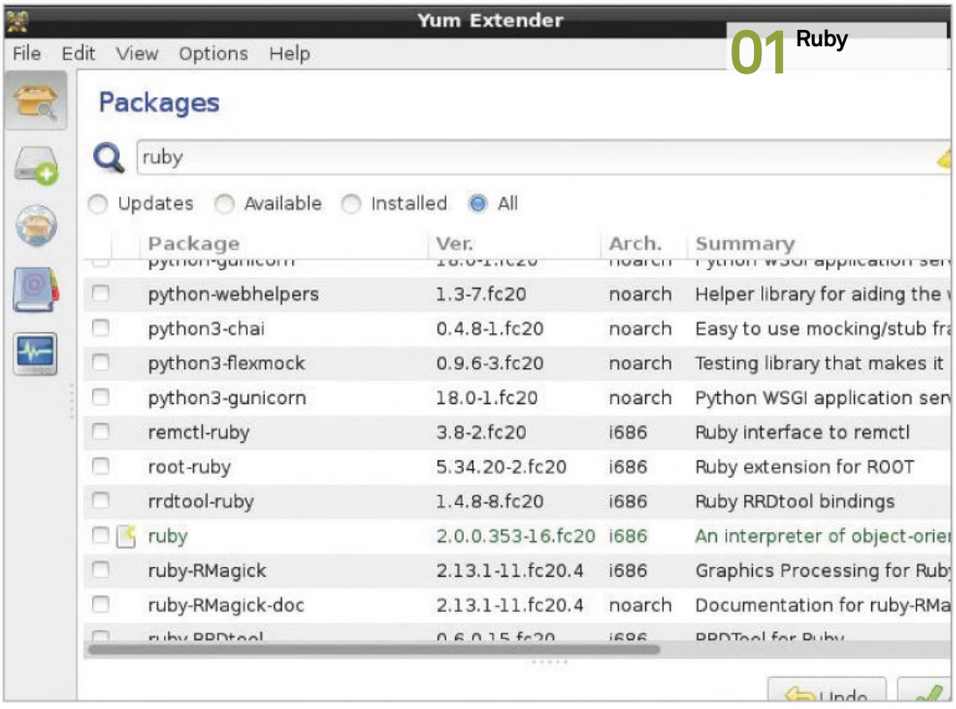
While Perl and Python are the scripting languages that Linux distros and packages have traditionally depended upon, Ruby is the first choice for much of the DevOps and modern Web dev environment, and it's Ruby you'll need for Puppet and Vagrant.

```
ruby -v
```

... will tell you what, if any, version of Ruby you have. You'll need at least 2.0 for these tutorials.

There are options like rbenv to maintain multiple versions of Ruby easily on your PC.





02 VirtualBox

If the problem with versions is that you're maintaining a piece of software needing an older version of Ruby, then provisioning a virtual machine to run both that environment and your app is a great reason for following the tutorial.

Now, while you've got the package manager open, install VirtualBox too.

Your distro may have split out several separate packages, like the GUI interface virtualbox-qt. Make sure you get the package with the kernel modules virtualbox-dkms and the headers or source for the kernel you're running, as well as VirtualBox itself.

03 Get the latest

Now for Vagrant we go straight to www.vagrantup.com/downloads – RPMs and Debs are available in 32- and 64-bit flavours, and your browser will probably prompt you to open your package manager when you download.

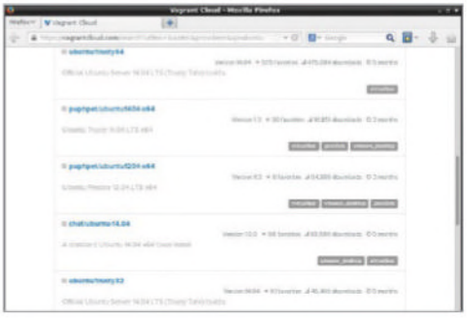
There's no need to call your package manager – install manually using, for example:

```
dpkg -i vagrant_1.6.5_i686.deb
```

... for the 32-bit package on Ubuntu or other Debian-based distros.

For other distros, download from GitHub and install with Rake, as outlined in the README on Vagrant creator Mitchell Hashimoto's GitHub page: https://github.com/mitchellh.

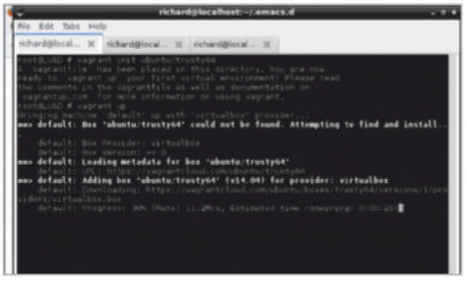
"Search VagrantCloud for lists of what you need in a setup"



04 Cloud-sourced

You'll need an OS image, and there are plenty available both at **www.vagrantcloud.com** and **www.vagrantbox.es.** You can search VagrantCloud for specific comma-separated lists of what you need in a setup such as jenkins,centos; wordpress,ubuntu; or rails,debian.

You'll find everything from minimal distros like Tiny Core (good for a quick download to test things out) to specialist, ready-rolled systems like data-science-toolbox/dst. For now, we'll stick to a basic setup of Ubuntu 14.04 – it's available from VirtualCloud in both 32- and 64-bit flavours.



05 Up and away!

Setting up a VirtualBox image from Vagrant is a simple matter of:

```
vagrant init ubuntu/trusty64
vagrant up
```

... which should download the Ubuntu 14.04 64-bit image from VagrantCloud and start it running. By default, the image should be kept in ~/VirtualBox VMs/ for subsequent use, but you can alter this in VirtualBox's preferences.

On most recent distros, everything should be hunky dory. But errors are not unknown, so we'll take a quick look at the most common problems.

06 Oops!

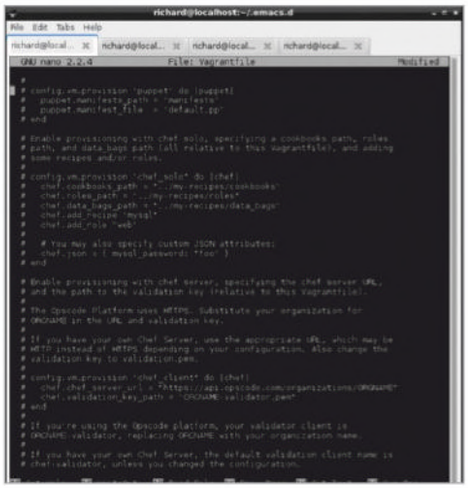
Problems? It's easy to miss the correct kernel headers during install. Check /proc /version (or run uname -a) to be sure which kernel you're running. Error messages from...

```
VBoxManage --version
```

... may help. On one Debian box, we had to rebuild virtualbox-dkms. For a Fedora test machine, we had to install the kmod-VirtualBox package for our kernel version, then run:

```
sudo systemctl restart systemd-modules-load.service
```

... which fixed the problem. You may find that a restart of your machine might be necessary for fixing problems.



07 Vagrantfile

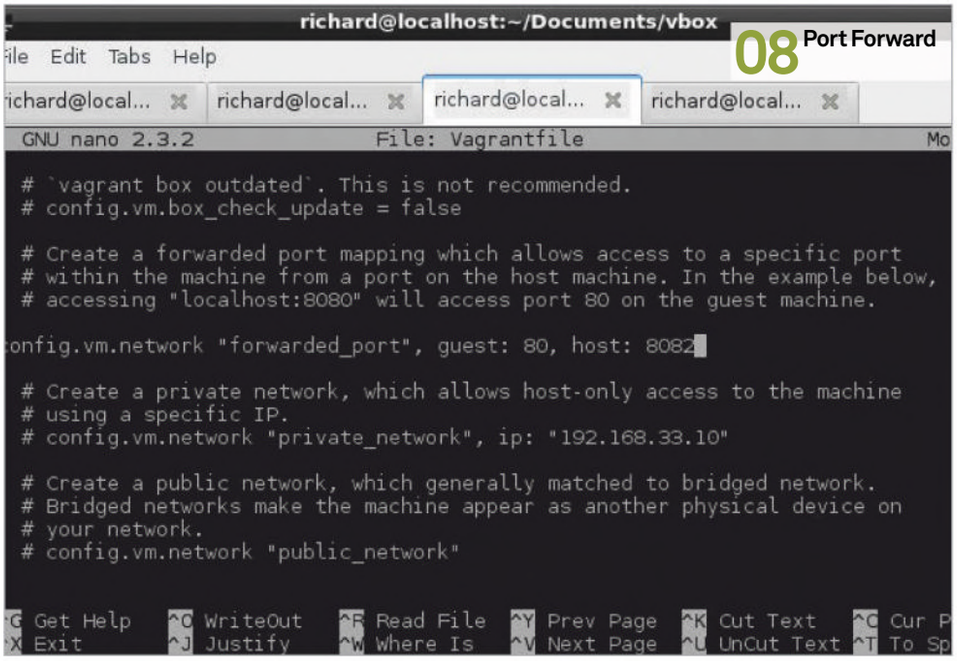
When you run vagrant init, you are told:

```
A `Vagrantfile` has been placed in this directory. You are now ready to `vagrant up` your first virtual environment! Please read the comments in the Vagrantfile as well as documentation on `vagrantup.com` for more information on using Vagrant.
```

Vagrantfile is where all of the configuration happens. Initially, everything is commented out save the config.vm.box value of ubuntu/trusty32 or whatever you set at vagrant init.

You can run vagrant init without a value and download the box you want later with the box add command. For example:

```
vagrant box add outnorth/debian-7.4RubyRailsDev
```



08 Port Forward

"The base image remains unaltered when it is used, so can be shared among several projects"

... then add it to the config.vm.box directive in Vagrantfile. Note that the base image downloaded remains unaltered when it is used, so can be shared among several projects – each one will have its own Vagrantfile in the local directory in which vagrant init was run.

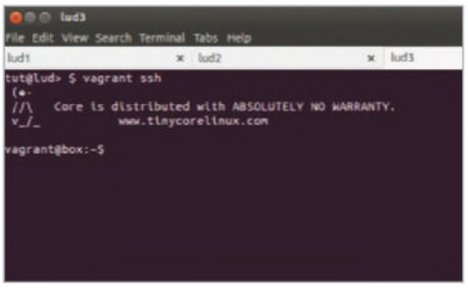
Whichever box you're running, setting up networking will be a necessity – you don't really want a website that can only be accessed from a local machine!

08 Port forward

In Vagrantfile, you can set a bridged network if that fits with your VM and hosting setup, but the simplest networking setup is port-forwarding. A port on your virtual box, such as 80, is forwarded to an unused port on your host machine, from where it can be accessed:

```
config.vm.network "forwarded_port", guest: 80, host: 8082
```

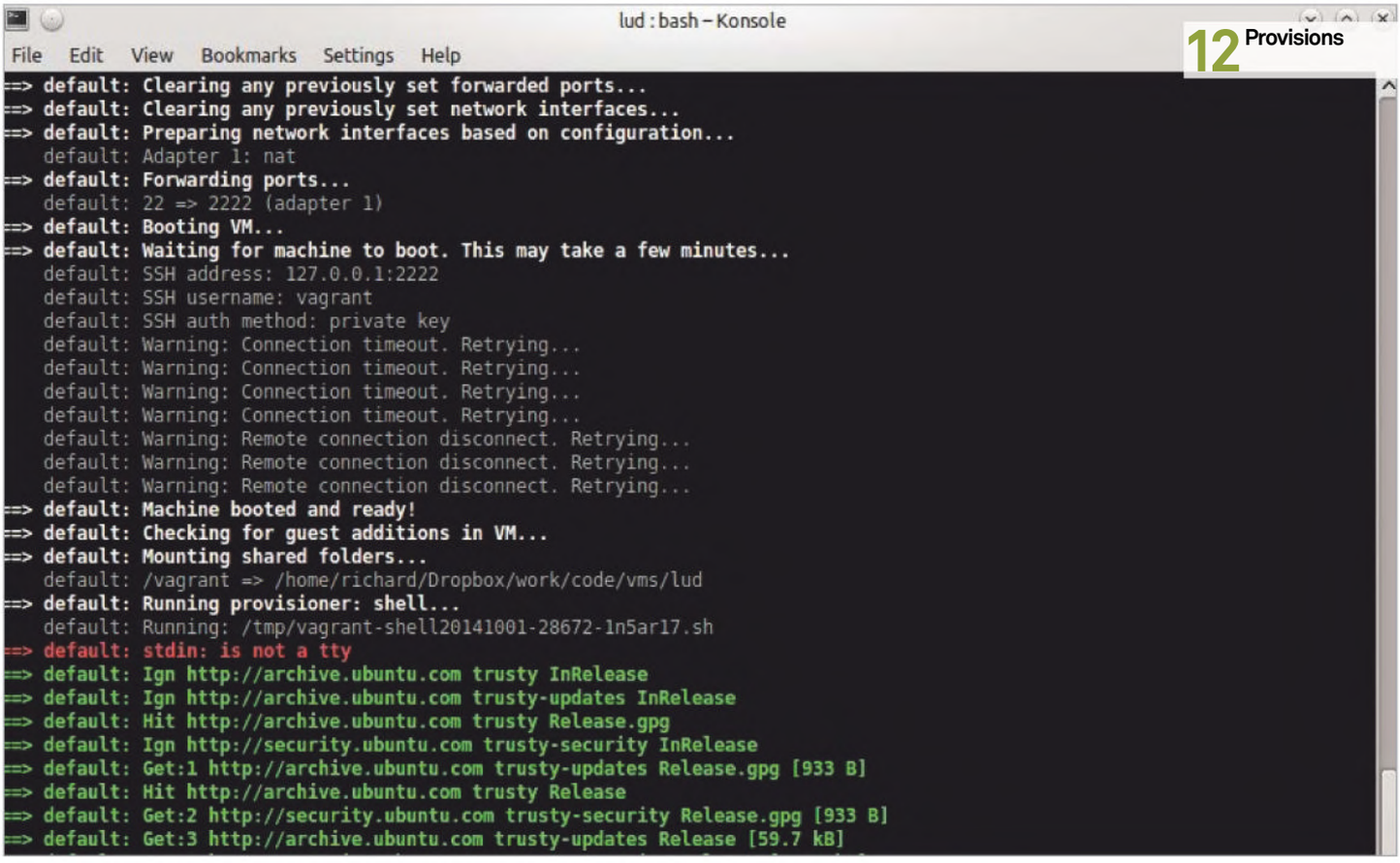
This in turn can be forwarded – for example, from Apache on the host machine – and/or matched there to the URL you want.



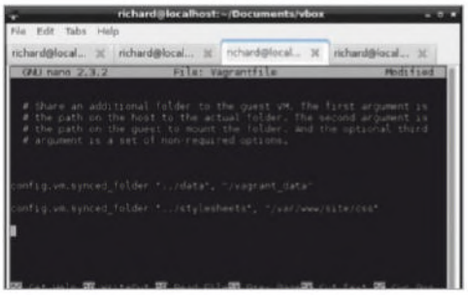
09 Shell

Changes to Vagrantfile can be applied to a running server with the vagrant reload command. While so much can be configured from outside your running server, vagrant ssh gives you all important access to the shell inside your virtual box Tiny Core, shown in the screenshot above, is great for quickly testing VBox, as opposed to using Ubuntu.

Don't forget to exit the ssh session before running any more vagrant commands. Vagrant suspend leaves the box a few seconds from readiness via another vagrant up. Vagrant destroy removes the virtual machine, but the Vagrantfile enables you to provision another that's exactly the same.



12 Provisions

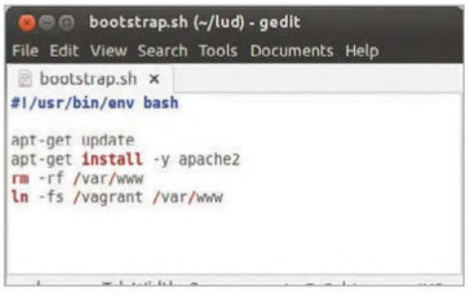


10 Shared files

Changes you make within a running box can be preserved; vagrant halt cleanly shuts down the box and saves disk contents. Added flexibility comes from being able to share files between the host and the virtual box.

By default, the directory from which you init the vagrant box is shared with that box. Take a look at /vagrant from within your ssh session – that Vagrantfile is the same one you were working on before.

More shared directories can be added by uncommenting config.vm.synced_folder in your Vagrantfile.



11 Bootstrap

Next month we're going to use Puppet to provision and maintain our virtual box, but we won't leave you hanging. Here's how to do it without Puppet, to get you going for now.

Create the file bootstrap.sh in the same directory as Vagrantfile. The canonical example (for a Debian or Ubuntu box) is:

```
#!/usr/bin/env bash

apt-get update
apt-get install -y apache2
rm -rf /var/www
ln -fs /vagrant /var/www
```

Note the linking of the web content to the directory shared outside the VM.

12 Provisions

The bootstrap.sh file is called by adding the following to Vagrantfile:

```
config.vm.provision :shell, path:
"bootstrap.sh"
```

... beneath the config.vm.box directive, and then using vagrant up --provision. Or, for an already created machine:

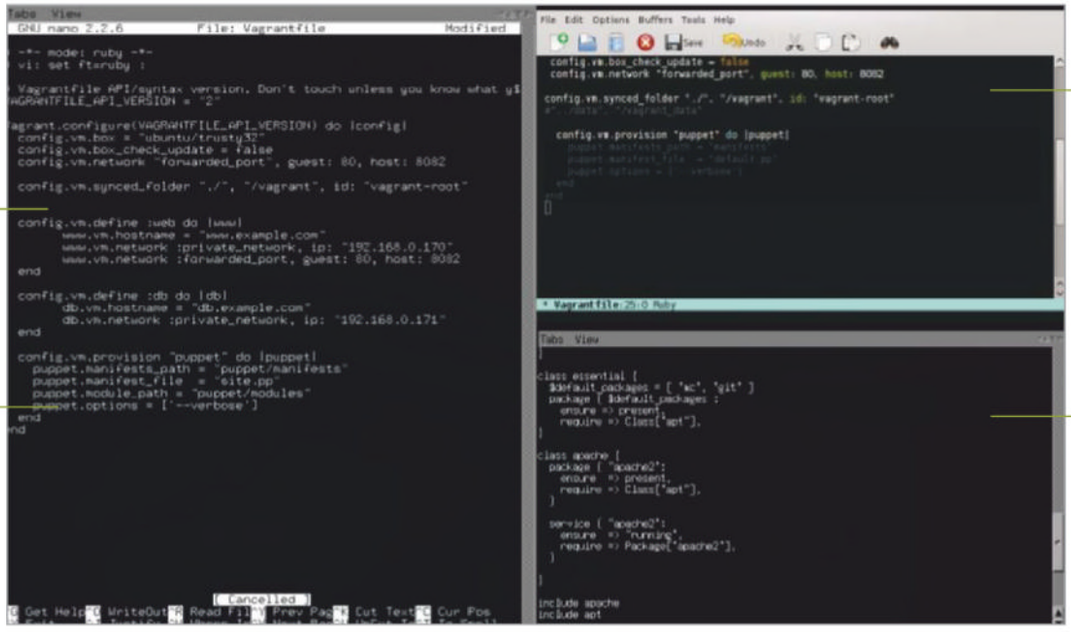
```
vagrant reload --provision
```

You'll see the output of the commands in bootstrap.sh on the terminal; expect a few warning messages but check through for anything unexpected.

Now, experiment with your bootstrap.sh file and perhaps different distro images. Over the page we'll show how as our needs grow more complex, Puppet keeps things maintainable.

Puppet keeps things maintainable and defining multiple nodes (VMs) is simplified

Vagrant is ready for Puppet – just uncomment the entries in Vagrantfile and the defaults will start you off nicely



Modules enable you to share reuseable chunks of Puppet config between projects

The Puppet manifest(s) can be simplified with classes, as a first step to modularity

Configure virtual boxes with Puppet and Vagrant – part 2

Previously, we used Vagrant to make deploying VMs simple. Now let's add Puppet to make complex deployments manageable

Advisor

Richard Smedley
A Unix jack-of-all-trades, Richard doesn't spend enough time in any language to get truly proficient, but always has a shell open so learnt scripting by osmosis

Resources

Puppet
Ruby ruby-lang.org
Virtual Box virtualbox.org
Vagrant vagrantup.com/downloads
Git (optional) git-scm.com

Back a few pages we saw how a simple virtual machine (VM) could be configured and then deployed anywhere simply by sending someone your Vagrantfile. We used a short shell script ('bootstrap.sh') to provision our example deployment with Apache. Sequential commands in a shell script, however, are not the most robust way to set up a VM, as they're hard to debug and to maintain as complexity grows.

Vagrant will work with other provisioners, including Ansible, Chef and Puppet. Ansible is a fair choice if you're starting a team afresh as it's easier to learn than the other two. Chef and Puppet have a steeper learning curve; Puppet and Chef's documentation can be confusing, so work through our tutorial to get the basics sorted, then head off in whatever provisioning direction your projects need.

Puppet will smooth over differences between package names on your CentOS and Debian VMs and handle provisioning multiple nodes simultaneously. Puppet's declarative style makes it a more natural fit for calling on for damage repair, and it handles complexity well with classes and modules. Its motto is "Automate everything".

Puppet is best learned by example, and we're going to be using it in a simple form, so you should be able to pick it up as you go along!

01 Masterful vs masterless

If you've used Puppet elsewhere, you'll be familiar with the Puppet Master. The nodes report to Puppet Master regularly and they provide central control of each node's configuration. It suits many use cases but not the largest cases (where companies with tens of thousands of nodes like PayPal apparently use Cron for the node reporting), nor for cases like ours where the Puppet manifest to provision the VM (one or only a handful of machines) is what we want.

Distribution is taken care of with Vagrant and your own deployment system (like Git or whatever else you use to distribute your development environments around your team). This also avoids problems with firewalls, scaling and the nontrivial matter of configuring the Puppet Master.

02 Repairable

That isn't to say that you'd never want to run a Puppet Master with Vagrant – it's just that it isn't necessary, and in most cases you'll manage very well without it. The Vagrant docs have a useful section on working with Puppet Master (see Puppet Agent, under the Provisioning section).

Our stand-alone Puppet not only still makes provisioning more reliable across different distros and versions of software, it also makes the running system repairable. Whereas the shell script we used last month can only set up the system when it's created, Puppet can compare the machine's actual state with what the manifest tells it should be happening, and repair the VM.

03 Something to declare

Our provisioning shell script was just a series of commands:

```
#!/usr/bin/env bash

apt-get update
apt-get install -y apache2
rm -rf /var/www
4ln -fs /vagrant /var/www
```

It at least provides a reproducible method that can be shared, but it doesn't travel across platforms. For example, if you want the same services on an Ubuntu and a CentOS server, you'll need scripts with different commands for each installation step.

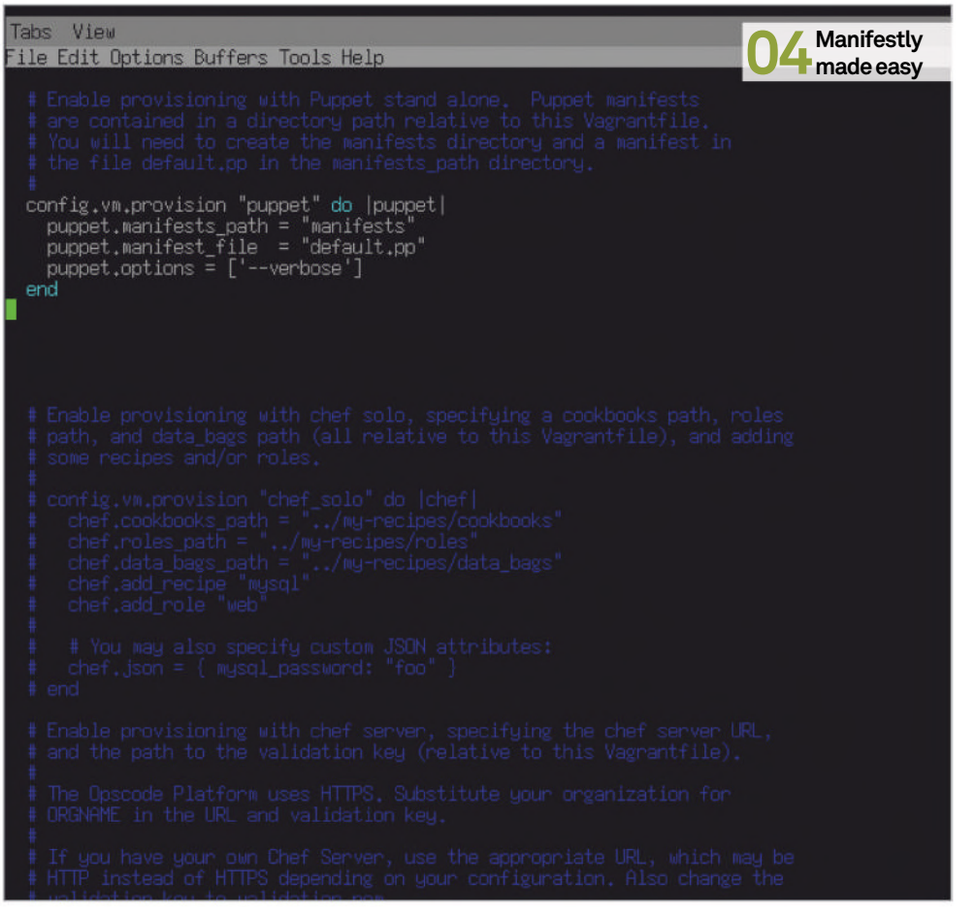
We'll start with the Puppet equivalent of that initial install script and build on that, examining the hows and whys as we go.

04 Manifestly made easy

We need to create a directory for manifest files, which tell Puppet how to provision the machine. Traditionally the directory is manifests/, which is in the same place as our Vagrantfile, and in which you'll refer to it by relative path. Large manifests can be split across files, possibly in a separate scripts directory, but a simple one can fit into the file 'default.pp'. In a fuller manifest this will just define the node(s) and basic parameters.

In our Vagrantfile we comment out or delete the reference to the shellscript we added last issue and instead uncomment the default Puppet options, adding in a --verbose switch to provide a bit more information in case of errors:

```
config.vm.provision "puppet" do |puppet|
  puppet.manifests_path = "manifests"
  puppet.manifest_file  = "default.pp"
  puppet.options = ['--verbose']
end
```



04 Manifestly made easy

"Simple manifests can fit into the file default.pp"

05 Require =>

To replace our basic shell provisioner, put the following in manifests/default.pp

```
exec { "apt-get update":
  path => "/usr/bin",
}
package { "apache2":
  ensure  => present,
  require => Exec["apt-get update"],
}
service { "apache2":
  ensure  => "running",
  require => Package["apache2""],
}
file { "/var/www":
  ensure => directory,
  owner => www-data,
}
```

```
Tabs  View
tut@lud > vagrant provision
==> default: Running provisioner: puppet...
==> default: Running Puppet with default.pp...
==> default: stdin: is not a tty
==> default: Notice: Compiled catalog for vagrant-ubuntu-trusty-32.home in environment production in 0.79 seconds
==> default: Info: Applying configuration version '1413505687'
==> default: Notice: /Stage[main]/Apt/Exec[apt-get update]/returns: executed successfully
==> default: Notice: Finished catalog run in 46.64 seconds
tut@lud > vagrant ssh
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-35-generic i686)

 * Documentation:  https://help.ubuntu.com/

  System information as of Fri Oct 17 00:27:55 UTC 2014

  System load:  0.11              Processes:           74
  Usage of /:   2.7% of 39.34GB   Users logged in:     0
  Memory usage: 16%               IP address for eth0: 10.0.2.15
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

47 packages can be updated.
23 updates are security updates.


Last login: Fri Oct 17 00:18:59 2014 from 10.0.2.2
vagrant@vagrant-ubuntu-trusty-32:~$
```

08 **Reprovision**

06 Manifestly apt

Instead of giving the steps (update the package list, then install Apache), we simply told Puppet that we required Apache to be present:

```
package { “apache2”:
  ensure => present,
}
```

Note that we had...

```
require => Exec[“apt-get update”],
```

...to tell Puppet that the system should be updated first. The distros' repositories usually contain updated (and more secure) versions of software than the ones shipping in release ISOs.

Once you've run `vagrant up` and `vagrant ssh`, take a look at /var/log/apt/history.log to check it ran, although you'll also see it working from the host by pointing your web browser at http://localhost:8082.

07 Class act

Beyond our simple case, requirements for packages and services soon get complicated. Organising them into classes makes files more maintainable, as requirements change, and gives you more reuseable components for where you have more than one node, such as for web servers, database servers and load balancers.

```
Exec { path => [ “/bin/”, “/sbin/” , “/usr/bin/”, “/usr/sbin/” ] }

class apt {
  exec { ‘apt-get update’:
    command => ‘apt-get update’,
  }
}
class apache {
  package { “apache2”:
    ensure  => present,
    require => Class[“apt”],
  }
  service { “apache2”:
    ensure  => “running”,
    require => Package[“apache2”],
  }
}
include apache
include apt
```

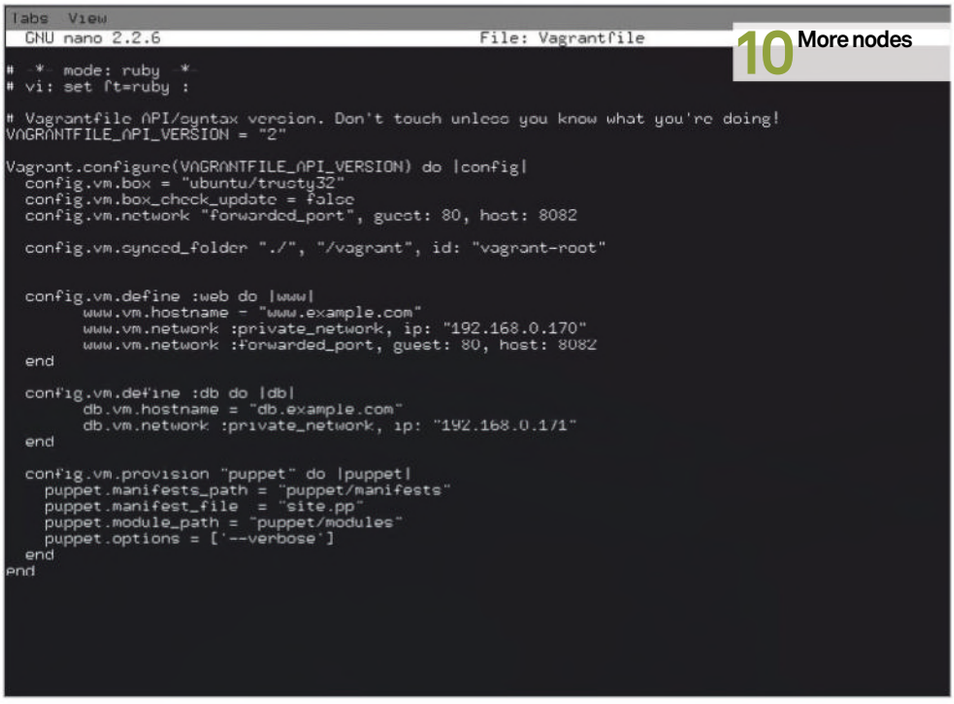
In our earlier manifest, we'd specified a path under the instruction to run apt-get update. Here we separate out the location of binaries to run into a path directive at the start. Everything else we can tidy into classes: we've just got apt and Apache for now, but feel free to add your own.

08 Reprovision

`vagrant provision` reloads the provisioner without rebooting the VM. If you've shut it down, `vagrant reload --provision` (or `vagrant up --provision` if starting again) makes sure the altered manifest is loaded. And so does this, from within the VM, via vagrant ssh:

```
puppet apply /vagrant/manifests/default.pp
```

Don't worry about time-outs when you're loading or reloading vagrant if there are no warnings. You'll see if it's all running okay anyway.



“Vagrant reload can make sure the altered manifest is loaded”

09 Clean and tidy

At this point you may want to get rid of all the commented-out code in Vagrantfile which you're not using and make a copy to keep handy for reference (though the default Vagrantfile is easily found again online). Snip out the unwanted sections on provisioning and networks that you don't need.

Now you can see the entire file in your text editor, it's easier to add in a few more things. We've put:

```
config.vm.box_check_update = false
```

...despite it being not generally recommended. If you're making several changes to your setup whilst on a slow internet connection, it's a handy time-saver. But just remember that you will need to comment it out again to bring your box up to date before you do anything important on it.

10 More nodes

Now let's get back to the Vagrantfile and have a quick look at providing more than one node (VM) from a single configuration. In Vagrantfile, before the provision directive:

```
config.vm.define :web do |www|
  www.vm.hostname = “www.example.com”
  www.vm.network :private_network, ip:
  “192.168.0.170”
  www.vm.network :forwarded_port,
  guest: 80, host: 8082
end
config.vm.define :db do |db|
  db.vm.hostname = “db.example.com”
  db.vm.network :private_network, ip:
  “192.168.0.171”
end
```

This defines two separate VMs – web and db.

11 Modular

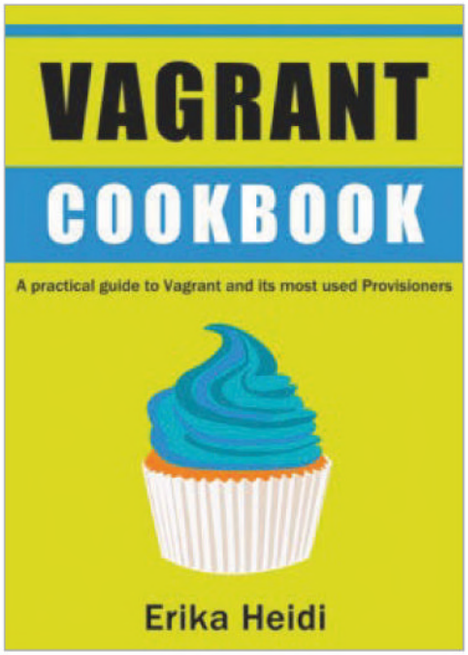
Now in the manifest file 'site.pp', we can split out the nodes with something like:

```
node www.example.com {
  include apache
}
```

There's plenty more to learn to really take advantage of Puppet here. Before we finish with a pointer to further resources, the last step to scaling and better maintainability is Puppet modules. A module groups resources in a directory: an 'init.pp' with a single class definition of the same name as the directory and module, plus other manifests as needed, to tidily contain the complexity of configuring resources like Apache.

You'll also need the path in your Vagrantfile:

```
puppet.module_path = “puppet/modules”
```



12 Moving on

We hope the diverse online documentation on Vagrant with Puppet will be more useful now that you have walked through a simple case with us. There's a lot of power in Puppet provisioning of Vagrant and plenty more tricks to try – such as applying version control. If you share a git repository with your development team, for example, then you have combined back up, version control and distribution.

In addition to the web resources that we've pointed to in the last couple of issues, John Arundel's Puppet 3 Cookbook (**bit.ly/1tySg95**) has detailed ideas on using Git to scale Puppet and is well worth a look. There's also a Vagrant cookbook based on the useful Vagrant blogs found at **erikaheidi.com**, and this book covers all of the provisioners.

Finally, try some ready-rolled VMs with Puppet examples to build upon – search VagrantCloud.com. Or just use **PuPHPet.com** to generate your manifests for you!

Build games for Pebble

Full code avaliable online: bit.ly/1toQAKs

Develop for the Pebble smartwatch and play games on the go

Sprites

We created custom artwork for our game

Background
Our background tile, which is repeated horizontally, is a 144 x 168 pixel contiguous design, meaning that it scrolls seamlessly as Tux moves around.

Objects
Our Tux character is a 30 x 30 pixel sprite. We also made some enemy sprites for the next part of this guide – check **FileSilo**.

Advisor

Tam Hanna has been in the IT business since the days of the Palm IIIc. Serving as journalist, tutor, speaker and author of scientific books, he has seen every aspect of the mobile market more than once

Under a superficial examination, the Pebble doesn't exactly look like an ideal device for gaming. If we look at its puny 80MHz processor, which is paired up with a monochrome e-paper screen, the display does not only lack the ability to display colour but is furthermore not particularly well suited to frequently updating content. The Pebble also interacts with its user via a grand total of four buttons, one of which cannot be accessed easily by third-party applications.

Paradoxically, the Pebble's success is directly rooted in its utilitarian approach to smartwatch design. Using low-powered hardware permits the watch to be cheap and cheerful: most of its competitors tip the scales far harder and therefore last for significantly less time on a single charge.

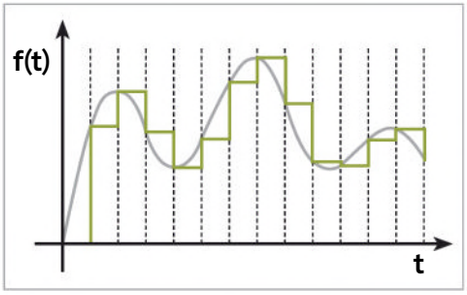
In the past, developers have frequently risen to the challenge of doing almost impossible things. Given that some gamers have managed to shoehorn a complete 3D engine into a Commodore 64, creating a game for one of the world's favourite smartwatches seems as though it should be more than doable.

Dong Nguyen's very popular game Flappy Bird recycled a simple game concept first seen on Palm OS. Originally known as SFCave, the game involves you leading an object through a cave where activating a booster rocket makes the object rise, while gravity makes it fall on its own.

Flappy Bird is primitive when analysed from a conceptual point of view, but this is beneficial for our cause; getting started with game programming becomes much easier if the example at hand does not distract you from the actual coding work.

The first – and utmost important – concept involves the idea of the game loop. GUI-driven applications spend most of their time gallivanting around and waiting for user input – as long as the user does not press a control, nothing has to be done. Input causes the app to perform more or less complex computations, after which the results are displayed.

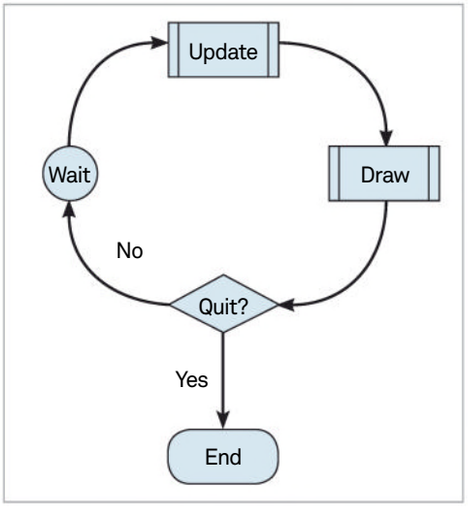
Games find themselves in a less satisfactory situation. They are to model the real world, which does not provide its subjects such a serene existence. Instead, everything is continuous: gravity pulls, people chatter and wines age in real time.



Discretisation transforms the continuous function into single-value steps

The first step to digital bliss involves a process called discretisation. This mathematically complex process splits real time into a group of slots (see graph above), which are then run one after the other. Inside each slot, all movements are considered discrete: the wine will age by one slot, while a person says a letter (or two).

When the individual steps are small enough, they cannot be taken apart from a normal and contiguous motion. In most cases, games will work with a loop-like structure similar to the one shown above: each update cycle is followed by one dedicated to refreshing the screen. The number of screen redraws



A game loop provides the framework for most animated visuals

is often referred to as the frame rate. Once rates reach more than about 30 frames per second, users tend to perceive them as continuous.

CODING ON THE INTERNET

Even though Pebble OS is supported by a Linux-based software development kit, most developers choose to do their work via CloudPebble. It provides a set of private repositories and a network-based compiler. Remote deployments are enabled via a smartphone connected to the Pebble.

Getting started with CloudPebble requires you to visit https://cloudpebble.net/ide with a browser of your choice. Proceed to creating an account in order to maintain your projects on the server. Your Pebble smartwatch must be connected to the Pebble Conduit app, which can be downloaded from the Google Play Store.

Once the initial configuration is done, click Menu>Settings and enable the developer connection. Then, open the menu in order to find the developer mode settings and check the Enabled checkbox in order to activate the forwarding (if your CloudPebble account matches the one used on the phone, the connection will be established automatically).

At the time of writing, developers need to use a beta version of the handset app because it only runs on Android 4.3 and above; the previously possible method of manual IP address entry has been disabled.

Next, you need to click on the Create button in order to start the New Project wizard. The Flappy Tux game is a Pebble C SDK Project based on the Minimal template. It consists of just the one file named main.c. Flappy Tux's default content looks like the following:

```
#include <pebble.h>

Window *my_window;
TextLayer *text_layer;

void handle_init(void) {
  my_window = window_create();

  text_layer = text_layer_create(GRect(0,
  0, 144, 20));
  window_stack_push(my_window, true);
}

void handle_deinit(void) {
  text_layer_destroy(text_layer);
  window_destroy(my_window);
}

int main(void) {
  handle_init();
  app_event_loop();
  handle_deinit();
}
```

Fig 01

```
void handle_init(void)
{
  my_window = window_create();

  myCanvas = layer_create(GRect(0, 0, 144, 168));

  window_stack_push(my_window, true);

  Layer* motherLayer=window_get_root_layer(my_window);
  layer_add_child(motherLayer, myCanvas);

  layer_set_update_proc(myCanvas, updateGame);
  app_timer_register(34, timer_handler, NULL);
}
```

"Normal applications tend to be made up of one or more windows cascaded above one another"

Main.c starts out by including pebble.h – this file contains definitions for the various operating system functions. We then proceed to create two pointers: one of them refers to a Window object, whereas the other pointer will address a TextLayer.

Pebble OS contains a layer-driven GUI stack. This means that normal applications tend to be made up of one or more windows cascaded above one another. Each of these window objects can contain one or more sublayers, which realise the actual user interface.

Game developers tend to shy away from the user interface resources provided by the OS due to their less than stellar performance. In the case of the Pebble, games based on multiple BitmapLayers tend to be significantly slower than ones based on the sprite drawing technique used in our example.

Handle_init and handle_deinit are responsible for setting up the user interface. It consists of an empty form containing an also empty label:

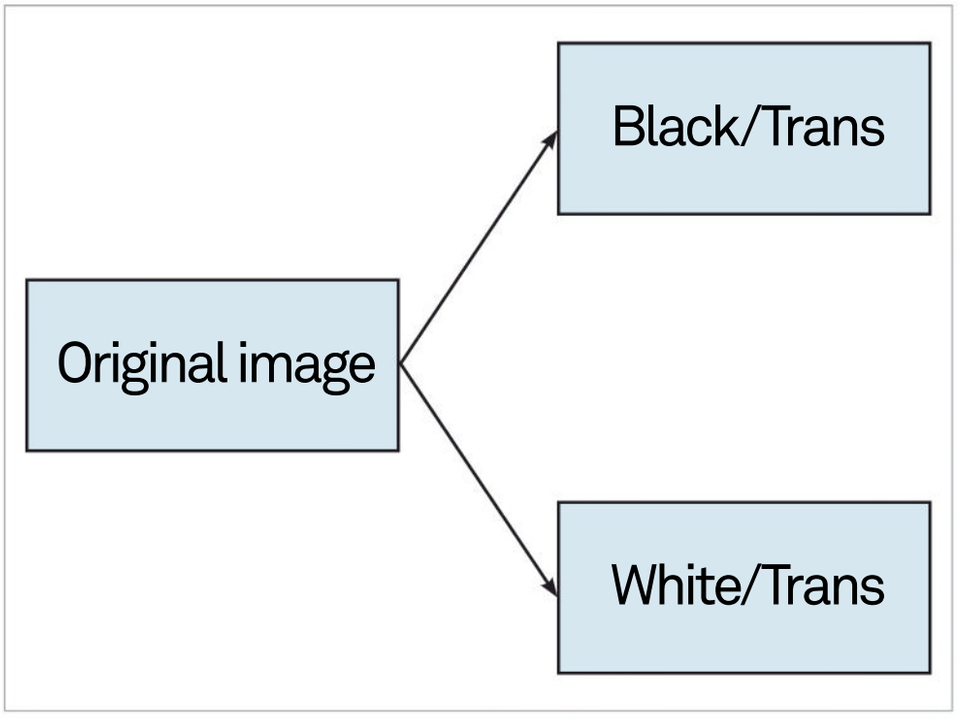
Pebble OS calls the main() function in order to start an application. It invokes the app_event_loop() method, which is responsible for starting the aforementioned event loop. The invocation will not return until the program is intended to terminate. Following on from that, handle_deinit() is invoked:

With that, we are ready to deploy our application to the smartwatch. Ensure that developer mode is enabled in the conduit app, then click the play button in the GUI of CloudPebble. An empty window should appear on your Pebble after a few seconds worth of waiting.

TICK, TICK, TICK

Our game will draw itself into a single layer. This requires us to start a game loop – a process that is ideally accomplished by changing the content of handle_init in order to look like the version shown in the listing in Fig 01 (previous page).

Handle_init now starts out by creating a fullscreen layer. It is then pushed into the main window in order



Discretisation transforms the continuous function into single-value steps

to make it appear on the display. Then layer_set_update_proc assigns an update handler to the layer.

Most operating systems don't permit developers to update the visuals as they please, but rendering can be made more effective if changes are made only in response to an event. Our method updateGame() will be invoked whenever myCanvas needs to be redrawn.

Since timers in Pebble OS are always single-shot, the timer handler must retrigger itself in order to keep the loop running. Furthermore, the layer is marked dirty in order to invoke its redrawing method:

```
static void timer_handler(void *context)
{
  layer_mark_dirty(myCanvas);
  app_timer_register(34, timer_handler,
                     NULL);
}
```

Finally, the actual redrawing will take place in updateGame. This method is provided with a GContext variable which will now point at the layer in question:

```
static void updateGame(Layer *layer,
                       GContext *ctx)
{

}
```

ADD SOME PHYSICS

Our version of Flappy Bird does not have to be full of features – we will, for now, be happy if our little Tux moves across the screen. This can be accomplished by 'integrating' the steering input in every pass of the game loop:

```
static void updateGame(Layer *layer,
GContext *ctx)
{
  totalPos+=moves_per_frame;
  moves_per_frame+=0.04;
  if(totalPos < 30) totalPos=30;
  if(totalPos > 114) totalPos=114;

  flownWay+=1;
```

Newtonian physics live by the law of constant motion. The position of an object can be determined by summing up its speed over time, and speed itself can be derived by summing up acceleration. Our example does this with two global variables, which are furthermore given a sanity check in order to keep Tux from falling off the screen.

SPRITES AND BITMAPS

Pebble OS provides developers with a group of graphical primitives that can be used for drawing lines, rectangles and circles. These methods can work really well when faced with simple tasks, especially because creating more complex visuals by hand imposes a significant performance penalty due to the complex mathematics required.

Ready-made elements can be brought on-screen much faster by using a prerendered bitmap. Displaying it involves but a few calls to memcpy. Our graphic artist has created a few ready-made PNG files for our image.

Click the Add New button next to the Resources header in order to open the resource adding wizard. The identifier field must be provided with a string that makes a valid C constant, which your code will use for finding the resource in question.

The background image is to be added as a normal PNG image. Due to the Pebble's lack of direct support for bitmasks, the character sprites need to be uploaded twice: both the 'black-centric' and the 'white-centric' versions should be transferred with a resource type of "PNG Image with transparency". This leads to a resource structure which is similar to the one shown on the left.

Any embedded resources must be decompressed before use. We accomplish this in handle_bmps, which is to be invoked from handle_init (Fig 02).

UpdateGame() must be expanded to include the drawing code responsible for handling the bitmaps (Fig 03).

Look carefully at our background tile. You will see that the contents of its right-most border now match the ones on the left: drawing more than one tile next to one another creates a continuous pattern.

We utilise this by drawing the background in two steps. Tile number one is drawn slanted to the left: the farther our character moves, the more of it gets drawn off-screen. The remaining white space is then filled with a second image, which partially overflows off the right-hand side of the display.

Drawing the actual sprite is a bit more involved. The calls to set_compositing_mode determine how the source image is to be rendered onto the underlying canvas. We start out by using Clear, which permits us to write the black parts of the Tux figure. After that, Or is used in order to paint the white eyes and belly of the penguin.

Compositing is quite a complex affair. You can find further information in the official documentation by looking up the term GcompOp at the Pebble developer site (**http://bit.ly/1vseKak**).

ADD INTERACTIVITY

Running the game as it stands will yield a moving background and a Tux falling down in a more or less naturally accelerated fashion. Take note to see that its belly is not transparent thanks to our expedient efforts.

Our Pebble has a total of four buttons. The back knob on the left-hand side of the watch closes the currently opened window; repurposing it requires significant effort, so we will restrain ourselves to handling pushes of the middle button. Sadly, handling knob events is quite a procedure under Pebble OS. Let's start the process by creating a configuration provider function, which will then be invoked from handle_init() (Fig 04).

Its raison d'être involves informing the system about your application's needs in relation to keyboard input. Our implementation registers interest in the central button, which will be routed to the two methods responsible for handling the clicking and unclicking of the button (Fig 05).

Our game loop analyses the flag. This information is used for determining the acceleration working on the penguin: if the button is held down, an imaginary booster cancels gravity and furthermore provides a slight jolt to the top:

```
static void updateGame(Layer *layer,
GContext *ctx)
{
  totalPos+=moves_per_frame;

  if(myIsClicked)
  {
    moves_per_frame-=0.06;
  }
  else
  {
    moves_per_frame+=0.04;
  }
  . . .
```

CONCLUSION

It's now time to call it quits for this issue of Linux User and Developer. We've managed to create the beginnings of a Flappy Bird clone, which further introduced us to a variety of interesting concepts related to game programming. Adding support for enemies and walls should be a breeze.

Things like game loops, sprite handling and basic physics are universal. The knowledge gathered here can be applied to games running on smartphones and PCs – even high-budget game titles such as Call of Duty are based on similar paradigms.

The next article will present an example wall implementation. It will then proceed to look at the second reason for the tremendous success of the Pebble: it can connect itself to the most commonly used smartphones and can display the content stored on them in a more convenient fashion.

We will exploit this by creating an Android-based conduit app that interacts with our smartwatch game.

"We've managed to create a Flappy Bird clone, which introduced us to a variety of interesting concepts"

Fig 02

```
void handle_bitmaps(void)
{
  myBG=gbitmap_create_with_resource(RESOURCE_ID_BG_SPRITE);
  myCharWhite=gbitmap_create_with_resource(RESOURCE_ID_CHAR_WHT_WHITE);
  myCharBlack=gbitmap_create_with_resource(RESOURCE_ID_CHAR_BLK_BLACK);
  myEnemyWhite=gbitmap_create_with_resource(RESOURCE_ID_WALL_WHT_WHITE);
  myEnemyBlack=gbitmap_create_with_resource(RESOURCE_ID_WALL_BLK_BLACK);
}
```

Fig 03

```
static void updateGame(Layer *layer, GContext *ctx)
{
  . . .

  graphics_context_set_compositing_mode(ctx, GCompOpAssign);
  graphics_draw_bitmap_in_rect(ctx, myBG, GRect(-flownWay%144, 0, 144, 159));
  graphics_draw_bitmap_in_rect(ctx, myBG, GRect(144-(flownWay%144), 0, 144, 159));

  graphics_context_set_compositing_mode(ctx, GCompOpClear);
  graphics_draw_bitmap_in_rect(ctx, myCharBlack, GRect(10, (int)totalPos, 20, 30));
  graphics_context_set_compositing_mode(ctx, GCompOpOr);
  graphics_draw_bitmap_in_rect(ctx, myCharWhite, GRect(10, (int)totalPos, 20, 30));

}
```

Fig 04

```
void config_provider(Window *window)
{
  window_raw_click_subscribe(BUTTON_ID_SELECT, sel_click_handler,
sel_release_handler, NULL);
}


void handle_init(void) {
  my_window = window_create();
  window_set_click_config_provider(my_window, (ClickConfigProvider)
config_provider);
  . . .
```

Fig 05

```
void sel_click_handler(ClickRecognizerRef recognizer, void *context)
{
  myIsClicked=true;
}

void sel_release_handler(ClickRecognizerRef recognizer, void
*context)
{
  myIsClicked=false;
}
```

Connect your Pebble game with Android

Pebble OS-based smart watches are social creatures. Let's teach Flappy Tux to get in touch with Android smartphones

Advisor

Tam Hanna has been in the IT business since the days of the Palm IIIc. Serving as journalist, tutor, speaker and author of scientific books, he has seen every aspect of the mobile market more than once

The 'evil' wall system draws the obstacles based on the player's position

The last tutorial you how to create a version of Flappy Bird for your Pebble smartwatch. Due to the complexity of the code, our game had to make do without a wall system: we simply didn't have enough space to go over its implementation.

Fortunately, this tutorial brings four new pages of Tam-generated goodness. In addition to a wall system, we will implement a form to display the current session's high score. Finally, a little conduit will be hacked up; it will connect Flappy Tux to your Android smartphone, opening up all kinds of fascinating possibilities for interaction.

Fine-tuning the difficulty of a game is one of the hardest design challenges. Simply increasing the speed of everything is easy – modifying the environment and/or artificial intelligence leads to more satisfying outcomes.

We use this approach by creating an 'evil' wall generation algorithm. Whenever a wall is created, the current position and speed of the player is taken into account. As we can compute the 'maximal' position which can be reached via Newtonian physics, a wall can be made higher or lower in order to modify the reaction time given.

GIVE ME… A PREDICTOR

New walls will appear on the right-hand side of the screen when the last wall is more than 60 pixels away. This means that the player has about 60 pixels worth of space in order to climb or sink. Since we assume a constantly accelerated motion, the distance travelled can be determined by the formula s = (0.5 * a * t2) + (v0 * t). 'v0' stands for the initial speed, while 'a' stands for the acceleration that is applied. In Pebble C, this can be implemented as in Fig 01.

getMaxClimb and getMaxDroop differ as they return the maximum value valid for raising and falling. Performing the power of two involves using mathematical functions, so its value is determined by transforming the operation into a series of multiplications: t2 becomes t*t (t3 would be t*t*t).

We use these values in order to create new walls. Wall positions get updated in the same pass, thereby creating an illusion of movement directed towards the player character:

```
static void checkWalls(GContext *ctx)
{
 if(wall1Alive==false && wall2x<60)
 {//Build a wall
   wall1x=140;

   wall1y= wall2y + getMaxDroop(60)/4;

   if(wall1y<20) wall1y=20;
   if(wall1y>100) wall1y=100;

   wall1Alive=true;
 }
 //Second wall omitted
 //Move walls
 wall1x--;
 wall2x--;
 if(wall1x==0) wall1Alive=0;
 if(wall2x==0) wall2Alive=0;
```

Walls must not be too close to one another. We accomplish this via mutual exclusivity: a new wall spawns only if the previous one has passed across half of the screen. A generous allowance is deducted to give the player ample reaction time. Furthermore, values are clamped in order to prevent the algorithm from going berserk.

Developers working on real games should offer different difficulty values. The actual amount of pixels to be subtracted should be determined by play testing. Coders working on existing games could also resort to analysing the behaviour of their current customers.

Drawing the actual walls is simple. We forego the use of sprites and render rectangles instead. You are, of course, free to change this if your application is to be released commercially (Fig 02).

graphics_fill_rect is interesting insofar as it permits the creation of rectangles with rounded

corners. This can be achieved by passing in a radius in lieu of zero; one or more GCorner flags can be ORred together in order to select the affected corners.

STYLISH COLLISIONS

Flappy Tux should display the player's current high score once the game has ended. CloudPebble's recently released GUI editor makes creating new forms really easy. Click the Add New button next to the Source Files section in order to open the creation wizard. Set File Type to Window Layout, and proceed to creating a new window called GameOverView.

The GUI editor is divided into two parts. Clicking control headers in the Toolkit section adds a corresponding widget to the form, while the properties of the currently selected widget can be modified in the aptly-named Properties pane. Add a group of controls in order to end up with the layout shown in the figure.

In the next step, proceed to clicking the ruler symbol on the right-hand side of your screen. CloudPebble will respond by showing you the generated code of the form, which will have a structure similar to Fig 03.

Qt developers will immediately recognise how the GUI editor works. The parts inside the UI comments are generated automatically whenever the layout of the form changes. Your code should confine itself to the methods outside of the comment – they are not regenerated as time passes by.

show_gameoverview activates the form for display. We can modify it in order to display the high and current scores, which are stored in global variables:

```
static char buf[] = "123456";
static char buf1[] = "123456";
void show_gameoverview(void)
{
 . . .

 snprintf(buf, sizeof(buf), "%d",
 highScore);
 text_layer_set_text(s_high, buf);

 snprintf(buf1, sizeof(buf1), "%d",
 myScore);
 text_layer_set_text(s_you, buf1);

 . . .
}
```

handle_window_unload gets called when form is removed from the screen. It is the ideal place to resume the game loop with a fresh start (Fig 04).

Fig 01

```
static float getMaxClimb(int t)
{
 return 0.5 * -0.06 * t * t + moves_per_frame * t;
}

static float getMaxDroop(int t)
{
  return 0.5 * 0.04 * t * t + moves_per_frame * t;
}
```

"Simply increasing the speed is easy – modifying the environment leads to more satisfying outcomes"

Fig 02

```
 //Draw wall
 if(wall1Alive)
 {
   graphics_fill_rect(ctx,GRect(wall1x, 0, 10, wall1y), 0, GCornerNone);
 }
 if(wall2Alive)
 {
   graphics_fill_rect(ctx,GRect(wall2x, 159-wall2y+32, 10, wall2y), 0, GCornerNone);
 }
}
```

Fig 03

```
// BEGIN AUTO-GENERATED UI CODE; DO NOT MODIFY
static TextLayer *s_you;
static TextLayer *s_high;
. . .
static void initialise_ui(void) {
. . .
}

static void destroy_ui(void) {
 . . .
}
// END AUTO-GENERATED UI CODE

static void handle_window_unload(Window* window) {
 destroy_ui();
}
. . .
```

Fig 04

```
static void handle_window_unload(Window* window) {
 destroy_ui();
 goverFlag=false;
 flownWay=0;
 totalPos=50;
 moves_per_frame=0;
 wall1Alive=wall2Alive=false;
 wall1x=wall1y=wall2x=wall2y=0;
 app_timer_register(34, timer_handler, NULL);
}
```

Fig 05

```
static void updateGame(Layer *layer, GContext *ctx)
{
 . . .
 checkWalls(ctx);
 if(wall1x<30 && wall1x>10 && wall1Alive==true)
 {
   if(totalPos<wall1y)
   {
     goverFlag=true;
     myScore=flownWay;
     if(highScore<myScore)highScore=myScore;
   }
 }
 //Second wall omitted
}
```

Fig 06

```
myConnected = PebbleKit.isWatchConnected(getApplicationContext());
if(myConnected)
{
       PebbleKit.startAppOnPebble(getApplicationContext(), PEBBLE_APP_UUID);
       PebbleDictionary data = new PebbleDictionary();
       data.addUint8(0, (byte) 1);

       PebbleKit.sendDataToPebble(getApplicationContext(), PEBBLE_APP_UUID, data);
}
```

Fig 07

```
Button aButton=(Button) findViewById(R.id.button1);
aButton.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v)
    {
        if(myConnected)
        {
            PebbleDictionary data = new PebbleDictionary();
            data.addUint8(0, (byte) 2);
            data.addInt16(1, (short)9000);
            PebbleKit.sendDataToPebble(getApplicationContext(), PEBBLE_APP_UUID, data);

        }
    }
    });
```

With that, the game loop must be updated one more time. Add the following snippet to the bottom in order to invoke the checkWalls function. Collisions with walls are handled by setting the game_over flag (Fig 05).

Our timer event handler does not preserve the AppTimer reference returned to it, which makes cancelling it a bit difficult. We solve this problem by parsing the GameOver-Flag in timer_handler:

```
void timer_handler(void *context)
{
 if(goverFlag==false)
 {
   layer_mark_dirty(myCanvas);
   app_timer_register(34, timer_handler,
                      NULL);
 }
 else
 {
   show_gameoverview();
 }
}
```

If goverFlag is set to true, no further frames are fed into the game engine. Instead, show_gameoverview is invoked in order to show the screen of doom.

ANDROID, AHOY!

Pebble supports Android and iOS. The Android SDK is available via a dedicated GitHub repository located at **https://github.com/pebble/pebble-android-sdk/releases**. We will use version 2.6 in the following steps, so simply click the link bearing its number to get it. Extract the archive and import the AndroidManifest file into Eclipse via Import>Android>Existing code. The SDK will show up as a project called main. Open the Properties dialog and navigate to the Android subsection: the checkbox "Is Library" must be enabled. Finally, drag and drop the contents of the java folder into src.

In the next step, the actual application is to be right-clicked. Open the Properties dialog and select the Android subsection. Click the Add button in the Library area. Eclipse will display a popup permitting you to select the "main" library created in the preceding step.

Pebble applications are identified via their globally unique UUID. Find yours in the Settings tab, and simplify access by creating a constant in your MainActivity:

```
public class MainActivity extends
ActionBarActivity {
    private final static UUID PEBBLE_APP_
    UUID = UUID.fromString("56f93cf8-1ab7-
    48c0-9859-d3c2f631c1db");
```

Pebble applications communicate with their companion applications via so-called dictionaries. A dictionary is best described as a key-value store – pass in an ID in order to retrieve the value associated with it.

For simplicity's sake, our Android conduit consists of one method. OnCreate starts by trying to find if a Pebble is currently connected to the smartphone. If that is the case, our application is brought to the foreground (Fig 06). A button must be pressed in order to transmit an artificial high score to the watch. Its implementation is interesting, mainly due to the way the OnClickListener is declared (Fig 07).

Both methods create an empty PebbleDictionary, which is then populated with one or more values. The individual tuples don't need to be stored in ascending order – a dictionary consisting of the values 1 and 50 would be perfectly legal.

Receiving information is a bit more difficult due to the way the interaction between watch and app is configured. PebbleKit is but a thin wrapper which fires intents into the driver, thereby saving your application from needing Bluetooth permissions. Harvesting data requires the use of a handler class. Our example combines this with a thread dispatch, which permits you to update the UI (Fig 08).

Pebble applications communicating via AppMessage should declare a total of four event handlers in main(). app_message_open informs

the operating system about the "chattivity" of your app, and furthermore permits you to specify the maximum size of incoming and outgoing dictionaries (Fig 09).

Space constraints force us to omit an explanation of the dropped/failed/sent handlers – their fairly primitive code can be seen in the example code on FileSilo.co.uk. inbox_received_callback is more interesting due to a unique constraint of Pebble OS – developers cannot provide a dictionary instance with an ID in order to receive the value in question. Instead, all tuples must be parsed one after another using a method like the one shown in Fig. 10.

Finally, the current highscore is sent to the smartphone via the timer_handler function. app_message_outbox_begin opens the outbox dictionary, which is then populated with the user data. app_message_outbox_send transmits the data to the smartphone, where it should be acknowledged by the client application:

```
void timer_handler(void *context)
{
 if(goverFlag==false)
 . . .
 else
 {
   DictionaryIterator *iter;
   app_message_outbox_begin (&iter);
   dict_write_int16(iter, 0, (int16_t)
   highScore);
   uint32_t final_size = dict_write_
   end(iter);
   app_message_outbox_send ();
   show_gameoverview();
 }
}
```

LEARN SOME MORE

Pebble OS is really not that difficult to work with; even after spending just a little time with it, you should find it easy to use. Developers who are used to classic PDA and smartphone OSs tend to be impressed by the simplicity of the API. Sadly, this does not mean that two tutorials of four pages each can cover the entirety of the features that are available to developers. Our treatment of the GUI stack is necessarily quite introductory.

However, Pebble itself has recently worked over its developer documentation. Open **developer.getpebble.com/** in your browser of choice in order to start learning more about what you can do; content found in Guides tends to provide more detailed information on specific topics, while the syntax and parameter roles of individual functions can be studied by selecting Ddocumentation.

Fig 08

```
final Handler handler = new Handler();
PebbleKit.registerReceivedDataHandler(this, new PebbleKit.
PebbleDataReceiver(PEBBLE_APP_UUID) {

 @Override
 public void receiveData(final Context context, final int
transactionId, final PebbleDictionary data)
 {
   handler.post(new Runnable() {
     @Override
     public void run() {
       TextView myView=(TextView)findViewById(R.id.textView1);
       myView.setText(String.valueOf(data.getInteger(0)));
     }
   });
   PebbleKit.sendAckToPebble(getApplicationContext(), transactionId);
 }
});
}
```

"Pebble OS is really not that difficult to work with; even after spending just a little time with it"

Fig 09

```
int main(void) {
 handle_init();
 app_message_register_inbox_received(inbox_received_callback);
 app_message_register_inbox_dropped(inbox_dropped_callback);
 app_message_register_outbox_failed(outbox_failed_callback);
 app_message_register_outbox_sent(outbox_sent_callback);
 app_message_open(app_message_inbox_size_maximum(), app_message_outbox_
             size_maximum());
 app_event_loop();
 handle_deinit();
}
```

Fig 10

```
static void inbox_received_callback(DictionaryIterator *iterator,
                                    void *context)
{
   Tuple *t = dict_read_first(iterator);
 while(t != NULL)
 {
   switch (t->key)
   {
     case 1:
       highScore = (int)t->value->int16;
       break;
   }
   // Get next pair, if any
   t = dict_read_next(iterator);
 }
}
```

Create your own VPN server

Dial into your own network from anywhere to access your files and browse the Internet freely

Advisor

Rob Zwetsloot models complex systems and is a web developer proficient in Python, Django and PHP. He loves to experiment with computing

Resources

OpenVPN openvpn.net

The use of VPNs is actually quite common around the world for a number of different reasons. Companies do it to enable their employees to dial into their office network and access relevant files and software. Individuals do it to dial into their own personal network for much the same reason, as well as being able to then use their own Internet connection to access online material that may be restricted if they happen to be using a hotel's Internet, for example.

In this tutorial, we are going to show you how to set up your own VPN server within your own network using the excellent OpenVPN software. As long as you have a system that you can keep up for 24 hours a day, this will be very useful for you. We are doing this tutorial on Ubuntu 14.04, but it will be easy to modify for most other systems too.

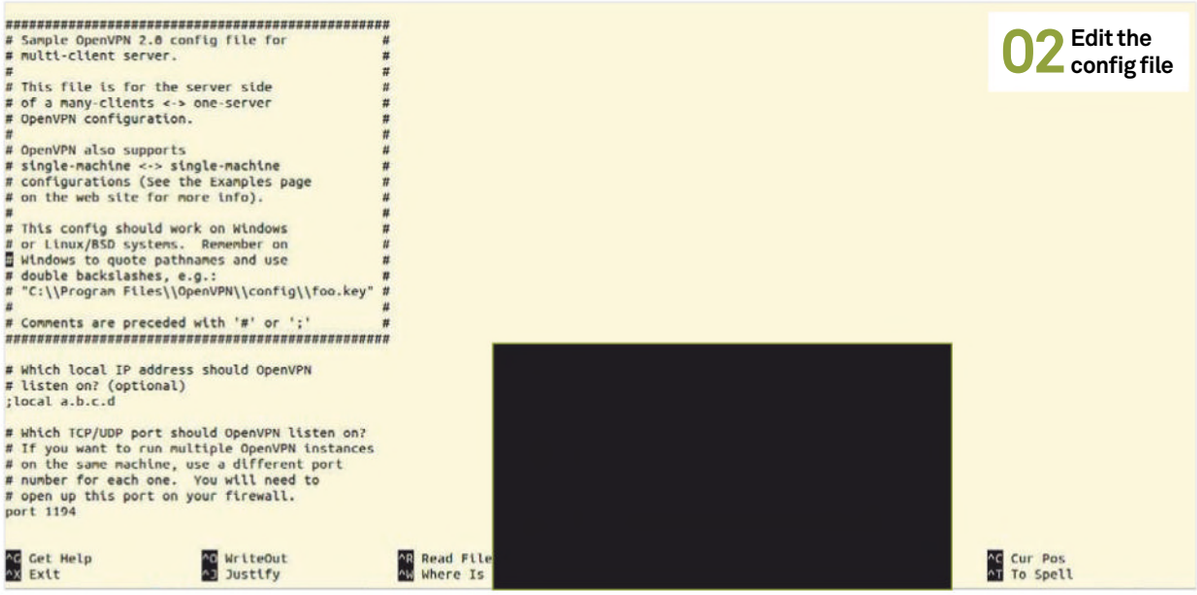
01 Initial setup

We're doing this on an Ubuntu machine, but everything we do will be translatable to other systems and servers. On your soon-to-be VPN server, you need to start by installing software with:

```
$ sudo apt-get install openvpn easy-rsa
```

Once that's done we need to get the example setup for us to work from and modify by doing:

```
$ sudo gunzip -c /usr/share/doc/openvpn/examples/
  sample-config-files/server.conf.gz > /etc/openvpn/server.conf
```



Left Instructions are found at the ends of the commented-out descriptions in server.conf

02 Edit the config file

We're going to start editing the config file example we just made by opening it first in nano (**sudo nano /etc/openvpn/server.conf**). Then change the following line:

```
'dh dh1024.pem' to 'dh dh2048.pem'
```

Remove the comment (;) from **;push "redirect-gateway def1 bypass-dhcp"**. Uncomment the lines below:

```
;push "dhcp-option DNS 208.67.222.222"
;push "dhcp-option DNS 208.67.220.220"
```

... and finally, also uncomment the following two lines before saving and exiting:

```
;user nobody
;group nogroup
```

03 Forward client internet

We can now edit sysctl to forward the packets from the computer that we are connecting from. We can do this by running the following command:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

We then need to edit the file sysctl.conf, so open it up in nano from the location /etc/sysctl.conf. Once open, we need to edit the line below:

```
#net.ipv4.ip_forward=1
```

... and remove the comment (#) so it looks like:

```
net.ipv4.ip_forward=1
```

Then save and exit.

04 An uncomplicated firewall

Uncomplicated firewall, or ufw, is installed by default in Ubuntu from 14.04 onwards and is as uncomplicated as its name suggests. We're going to allow OpenVPN to connect to and through it using the following two commands:

```
# ufw allow ssh
# ufw allow 1194/udp
```

Once those rules have been written, open up the ufw config file with nano at /etc/default/ufw and change DROP to ACCEPT in the following line:

```
DEFAULT_FORWARD_POLICY="DROP"
```

05 Make the rules

We need to make some new rules for the way the network address is translated and the way IP is masqueraded. To do this, we need to open up before.rules using nano at the location /etc/ufw/before.rules, and then add the following after the first paragraph:

```
# START OPENVPN RULES
# NAT table rules
*nat
:POSTROUTING ACCEPT [0:0]
# Allow traffic from OpenVPN client to eth0
-A POSTROUTING -s 10.8.0.0/8 -o eth0 -j MASQUERADE
COMMIT
# END OPENVPN RULES
```

06 Enable the firewall

Once you've saved these new settings you can finally enable ufw for use. To do this in the terminal you'll want to type:

> **"We're doing this on an Ubuntu machine, but everything we do will be translatable to other systems"**

Possible clients

You can also use OpenVPN to connect to the VPN server, which is a lot easier on Linux systems than some others as it is usually fairly quick to set up once the server is done. Remember that you will always need to create a profile and key for each client.

Create new keys

If you want to keep creating keys for different clients then you can just repeat Step 15 and give a different name to the client. Try to keep the names different so that you won't get any clashes occurring within the system.

```
# ufw enable
```

It will ask you if you really want to enable ufw, to which you can type 'y' to confirm. It will inherit all of the settings we've just modified and added, which will ultimately enable us to reroute traffic from a client system.

07 Check the firewall status

You can check the status of the firewall using **ufw status** at any time – it will let you know whether it's on or not and what rules and actions are currently being taken on certain ports. It only shows what the ports are doing when on, though.

08 Install Certificate Authority

We need to be able to have a way to generate and authenticate certificates for both the server and client to enable for secure communication between the two. To do this, set up a Certificate Authority. Start by copying the RSA scripts we looked at over with:

```
# cp -r /usr/share/easy-rsa/ /etc/openvpn
```

... and then create the following directory for key storage: /etc/openvpn/easy-rsa/keys.

09 Easy variables

We need to change the variables for RSA so we can create keys for our specific purposes and location. To do this open up the config file / etc/openvpn/easy-rsa/vars with nano and edit the following to your spec:

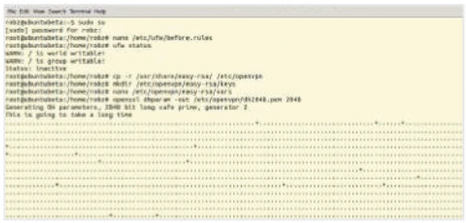
```
export KEY_COUNTRY="UK"
export KEY_PROVINCE="DT"
export KEY_CITY="Bournemouth"
export KEY_ORG="Amazing, Inc"
export KEY_EMAIL="rob@amazing.com"
export KEY_OU="Department of Excellence"
```

10 Name the server

The final part of the vars file to edit is the 'export KEY_NAME' line below these details. For the sake of making this tutorial easy, change it to:

```
export KEY_NAME="server"
```

If you wish to give it a different name, you'll have to start editing the OpenVPN config files that reference server.key and server.crt. For now then, we will refer to it as server and you can change it if and as you wish.



11 Generate parameters

We need to generate a Diffie-Hellman parameter for the key exchangce – this is a secure way of exchanging encrypted keys over a public network and essential for our server. To do this, you'll need to type the following command into the terminal:

```
# openssl dhparam -out /etc/openvpn/dh2048.
pem 2048
```

It takes a while as it calculates everything that is needed.

12 Build Certificate Authority

We need to do a bit more prep first before we build our CA. Firstly, to make things easier use **cd** to move into the /etc/openvpn/easy-rsa. From here you can initialise the public key infrastructure using the command below:

```
# . ./vars
```

It will warn you that by running clean-all you will be deleting all the current keys. As they're old and we don't run them, we are going to do exactly that:

```
# ./clean-all
```

Finally, actually build the CA using:

```
# ./build-ca
```

13 Server keys

Now we are going to build a security key for our VPN server. You can do this in the terminal from the same directory as before by using:

```
# ./build-key-server server
```

It will ask you again to confirm the details of the location of the server, but it will also ask you to add a password. Leave the password fields blank. You do want to sign and commit the certificate, so hit 'y' on those.

14 Turn OpenVPN on

We now need to move the keys to a location that OpenVPN actually expects to see them; in this case it's /etc/openvpn. We need to do the move using:

```
# cp /etc/openvpn/easy-rsa/keys/{server.
crt,server.
  key,ca.crt} /etc/openvpn
```

Once that's done, you can finally start the server ready to receive clients by using:

```
# service openvpn start
```

Use **service openvpn status** to check if it's on or not.

15 Create a key for the client

We can now build a unique key for the client that is based partly on the name of the client itself. You can create it from the directory we are already in by using:

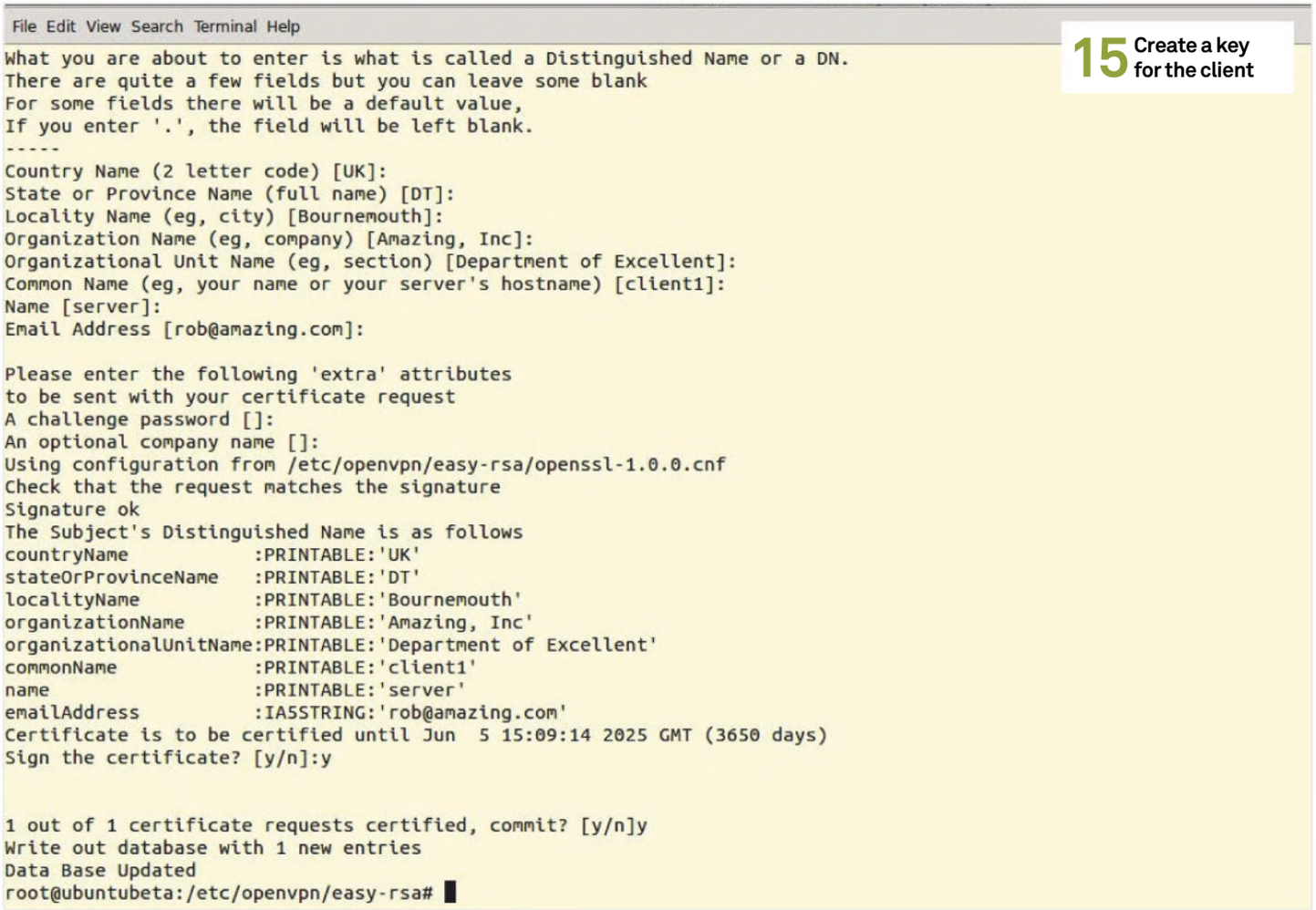
```
# ./build-key [name]
```

Again, don't give it a password or optional company info, and make sure you agree to sign the certificate and commit it.

16 Prepare the client configuration

The example file that we can use on all the clients is already in the filesystem, but we need to move it to the right location for us to make the make the most of it. We're also going to change the filetype of the example during the copy to better suit what the client will expect:

```
# cp /usr/share/doc/openvpn/examples/
sample-
  config-files/client.conf /etc/openvpn/easy-
rsa/
  keys/client.ovpn
```



15 Create a key for the client

```
File  Edit  View  Search  Terminal  Help
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [UK]:
State or Province Name (full name) [DT]:
Locality Name (eg, city) [Bournemouth]:
Organization Name (eg, company) [Amazing, Inc]:
Organizational Unit Name (eg, section) [Department of Excellent]:
Common Name (eg, your name or your server's hostname) [client1]:
Name [server]:
Email Address [rob@amazing.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/openvpn/easy-rsa/openssl-1.0.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'UK'
stateOrProvinceName   :PRINTABLE:'DT'
localityName          :PRINTABLE:'Bournemouth'
organizationName      :PRINTABLE:'Amazing, Inc'
organizationalUnitName:PRINTABLE:'Department of Excellent'
commonName            :PRINTABLE:'client1'
name                  :PRINTABLE:'server'
emailAddress          :IA5STRING:'rob@amazing.com'
Certificate is to be certified until Jun  5 15:09:14 2025 GMT (3650 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
root@ubuntubeta:/etc/openvpn/easy-rsa#
```

"Once you've set up a way to connect to the server from clients, you can begin testing the server out"

17 **Move files to client**

Once all the keys and example files are set up, you need to move certain files to whatever client you want to use to connect to this server. This includes four specific files: the first two are client-specific and use the name that we specified earlier as follows:

```
/etc/openvpn/easy-rsa/keys/[name].crt
/etc/openvpn/easy-rsa/keys/[name].key
```

The other two files are used on every client, and they are the following:

```
/etc/openvpn/easy-rsa/keys/client.ovpn
/etc/openvpn/ca.crt
```

Do this with all the clients.

18 **Connect remotely**

Once you've set up a way to connect to the server from clients, you can begin testing the server out and using the full facilities of a VPN. Whether you're doing it for business or just at home, it's an excellent way to work or use the Internet in an unrestricted way.

100,000 Stars is an interactive Google Chrome experiment, using WebGL to create impressive visuals detailing our nearest stars

Render 2D and 3D graphics with WebGL

Master graphics in your browser by learning to write WebGL programs to display 2D and 3D objects with JavaScript

Advisor

Mihalis Tsoukalos
is a UNIX administrator, a programmer (UNIX and iOS), a DBA and a mathematician. He has been using Linux since 1993

Resources

OpenGL opengl.org
OpenGL ES khronos.org/opengles
WebGL khronos.org/webgl
Reference Cards bit.ly/1zcQ8a4

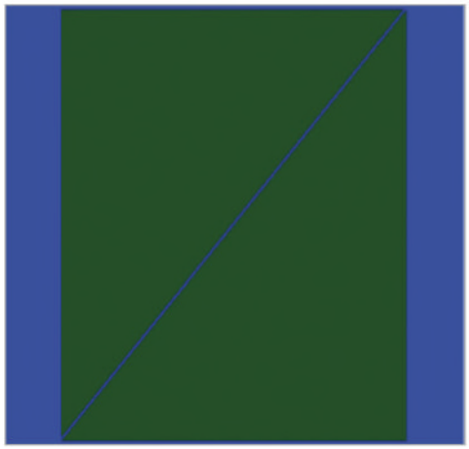
OpenGL is a well-known standard for generating 3D as well as 2D graphics; it's extremely powerful and has many capabilities. OpenGL is defined and released by the OpenGL Architecture Review Board (ARB) and is a big state machine. Most calls to OpenGL functions modify a global state that you cannot directly access. WebGL is a JavaScript implementation of OpenGL ES 2.0 that runs on the latest browsers. The OpenGL ES (Embedded Subsystem) is the mobile version of the OpenGL standard and is targeted towards embedded devices. OpenGL ES is a C-based, Platform-Neutral API. The OS must provide a rendering context that accepts commands as well as a framebuffer that keeps the results of the drawing commands.

All modern web browsers support the WebGL API by default so you do not need to do any extra work to use WebGL. If you want your WebGL programs to be available to the world, you will need to put them in a web server. Finally, all the code referred to in the steps can be downloaded from **FileSilo.co.uk**.

01 How to use WebGL

Before starting to write small WebGL programs, you will need to write some HTML code to use it as a template. The HTML code itself does not really matter in this instance as long as it fits your needs (see 'init.html').

The code does nothing but initialise some basic stuff that you should use later. The most important stuff is the definition of canvas; everything will be drawn inside the canvas element. You will also need to use a context name: some supported context names are 'webgl', 'experimental-webgl' and 'webkit-3d'. You can paint the canvas any colour you want.



02 Draw a simple shape

Drawing 2D or 3D shapes used to require expensive hardware and specialised software but nowadays, even a smartphone can run WebGL.

The presented HTML code (see 'rectangle.html') draws a 2D rectangle on-screen using WebGL. In order to separate the two triangles that compose the rectangle, the coordinates for the second triangle are slightly changed in order for you to distinguish the two triangles.

If you don't want a line to separate the two triangles, you should define the vertices array using the following code:

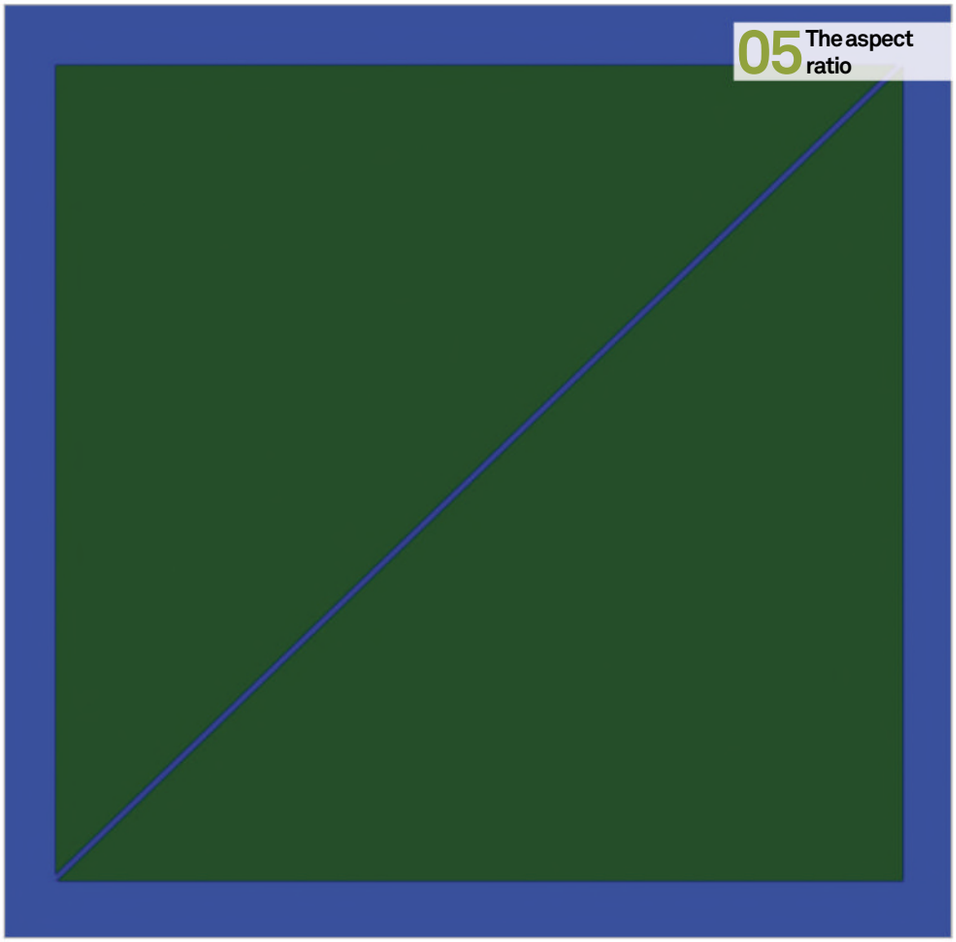
```
var vertices = [-0.75, -0.75, 0.75, -0.75, 0.75,
0.75, -0.75, -0.75, 0.75, 0.75, -0.75, 0.75 ];
```

03 The code explained

The JavaScript code is pretty straightforward. Nevertheless, it contains some mathematics that is needed for defining the coordinates of the rectangle's vertices.

The following line defines a buffer which will hold the vertex data:

```
var myVertexBuffer = gl.createBuffer();
```



The following line of code defines the number of triangles you want to draw:

```
gl.drawArrays(gl.TRIANGLES, 0, 6);
```

'createShader(str, type)' and the 'createProgram(vstr, fstr)' functions are standard code that you are going to use in your programs in some form or another.

Please also note that it would be a good idea to specify the size of the canvas that you are drawing onto using pixels.

04 About drawing

In WebGL there are three types of drawing primitives: points, lines and triangles. The most widely used primitive is the triangle, as every 3D object in WebGL is composed of triangles. The programmer should give the coordinates of the triangles that compose the desired shape. For drawing a rectangle you will need to draw two triangles. Therefore, you will need an array with six points in 2D dimensions because each triangle needs three points in order to be defined.

05 The aspect ratio

The programmer has to deal with the proportions of the WebGL native coordinate system. So, in order to make a square look like a square on-screen, you will have to calculate the aspect ratio. Then, you multiply the y values of each vertex by the aspect ratio and you are done.

Be careful of confusing the WebGL native coordinate system with the Cartesian coordinate system that is used for specifying the points in the programs you are using.

Drawing the rectangle using the correct aspect ratio requires the following changes to the code:

```
var AR = myCanvas.width / myCanvas.height;
var vertices = [-0.75, -0.75 * AR , 0.75, -0.75
* AR , 0.75, 0.75 * AR , -0.75, -0.74 * AR ,
0.74, 0.75 * AR , -0.75, 0.75 * AR];
```

You can see from the output that the canvas has the same size as before, but the shape is now sketched as a square (see 'aspectRatio.html').

06 The gl.drawArrays function

The WebGL function used for drawing is called drawArrays.

The first argument to the gl.drawArrays function specifies the drawing mode. If you want to draw a solid triangle, you will use gl.TRIANGLES. Alternatively, you can use gl.LINES and gl.POINTS for drawing lines and points respectively. The third argument of the function is really important because it declares the number of points that you will get as input. When you have problems with your output, check this parameter first.

07 Draw multiple shapes on-screen

This part will show you how to draw multiple rectangles and triangles on-screen (see 'multipleShapes.html'). Each shape will have its own colour. This method is particularly useful as it shows how to create autonomous functions that generate objects based on user-defined parameters. This means that you can create your own JavaScript library of useful functions that you can reuse. The secret is that you fill the WebGL buffer with data – that is, the shapes you want – until you finish with your drawing and then you display the buffer on screen!

08 Explaining the code

The program uses an external JavaScript file with various help functions called 'utils.js'. Its purpose is to avoid writing the same functions inside your scripts every time. Another important reason for using an external file is that if you find a bug or you make an improvement to an existing function, you only have to make changes to one file.

The two shaders are now stored in two separate <script> tags and compiled using the following code:

```
vertexShader = createShaderFromScriptElem
ent(gl, "2d-vertex-shader");
fragmentShader = createShaderFromScriptEl
ement(gl, "2d-fragment-shader");
program = createProgram(gl, [vertexShader,
fragmentShader]);
gl.useProgram(program);
```

It is not necessary to understand every single line of JavaScript code in order to experiment with WebGL. Use the existing code as a template, program your own draw functions and create your own shapes!

09 Using four triangles to create a rectangle

Now let's look at using more triangles to construct a rectangle (see 'rect4tria.html'). Although using more than two triangles for a rectangle is redundant, there are cases where you will need to describe a shape using more triangles to generate a smoother and more accurate shape.

To better understand the decomposition, each rectangle will use a different colour. Because the point with coordinates (200, 200) is a common point for all triangles, every triangle uses the (200, 200) point. As triangles need just three points to be defined, the actual order of the points is unimportant. The point order is more important for shapes with more than three points.

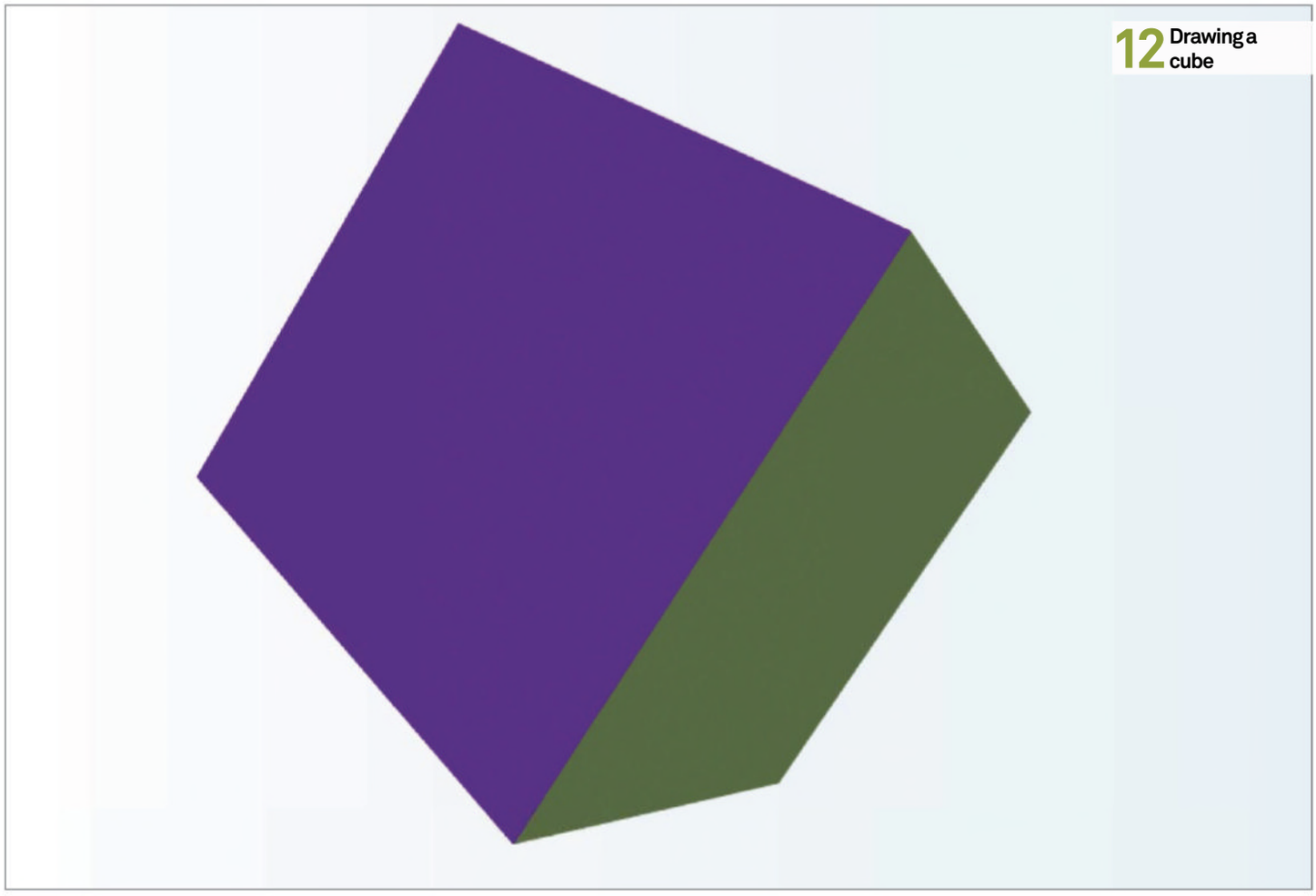
10 About shaders

There are two kinds of shaders: vertex and fragment. The vertex shader gets executed first and the fragment shader gets executed second.

Vertex shaders are used for controlling the points or vertices on a shape. In order to rotate an object in space, your vertex shader is responsible for applying a user-defined matrix to rotate your coordinates.

Fragment shaders are used for defining the colour, texture mapping, lighting and depth values for each pixel. The gl_Position variable defines which pixel of the screen to draw on, while the gl_FragColor variable defines the colour it should be.

The programming language used for defining shaders looks a lot like C. You can even write your shader code using separate files and include it in your JavaScript code. Both kinds of shaders run entirely on the GPU of the graphics card; therefore you want to keep the GPU as busy as possible in order to let the CPU do the other work.



11 The WebGL Pipeline

The programmable rendering pipeline makes it possible to write your own functions to control how shapes and images are rendered with the help of vertex and fragment shaders. There's a good digram of the rendering pipeline in Dev.Opera's Raw WebGL 101 guide: **bit.ly/ZWgkFW**. The pipeline is complex and shows how WebGL works behind the surface. Note that the vertex and fragment shaders are compiled and linked before used.

12 Drawing a cube

There is a JavaScript library called Three.js (**threejs.org**) that makes WebGL very easy to use by allowing you to create WebGL programs using less code. Its disadvantage is that the code is not directly portable to either OpenGL or OpenGL ES. Its advantage is that you can create sophisticated programs without having to write many lines of JavaScript code.

Our example (see 'cube.html') creates and rotates a cube in a 3D space. Writing a similar program in plain WebGL JavaScript would require over 500 lines of code and an article of its own!

13 Explaining the code

The Three.js library is high level compared to WebGL. You do not need to write code to initialise all things. You create a cube using just one line of code:

```
var cube = new THREE.Mesh(new THREE.
CubeGeometry(250, 250, 250),
new THREE.MeshFaceMaterial(materials));
```

You can use different colours for each cube face using the optional variable materials. You then need to add the cube to the scene in order to be displayed using the following line of code:

```
scene.add(cube);
```

14 Final words

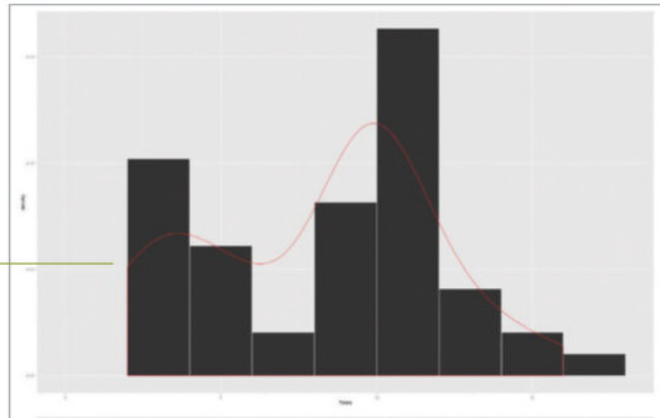
If you were going to construct the cube using WebGL and triangles, you would need to define its six faces. Each face needs at least two triangles to be defined. The reason I am saying at least is that, as you saw before, you can use more triangles if you want but you are going to need 12 triangles in total.

The main advantages of WebGL are that it is easy to learn if you are already familiar with either OpenGL or OpenGL ES, its code is directly portable to both OpenGL and OpenGL ES and that you have greater control over what you are drawing.

If you want to do something fast, then learn Three.js, but if you are more concerned with having full control over what you are doing, then WebGL is the way to go!

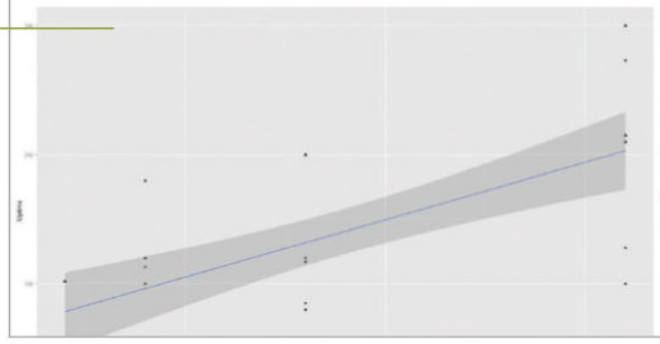
"If you want to do something fast, then learn Three.js"

This is a density plot that is drawn on top of a histogram. Drawing using layers has many advantages



This is the full R code of the 'chrome.R' script used in step 11. The produced image is quite simply amazing

This plot combines a scatter plot layer with a smooth layer. The shape of each point depends on the value of the Linux variable found in the LUD dataset



This is the format and some of the data from the LUD dataset used in this article for illustrating the varied capabilities of ggplot2

	RAM	Linux	Years	SSD	Uptime
M1	8	YES	9	128	100
M2	4	YES	5	256	102
M3	16	NO	3	512	80
M4	32	YES	10	1024	300
M5	8	YES	10	256	120
M6	32	YES	10	1024	210
M7	16	NO	3	512	117
M8	32	YES	2	1024	215
M9	16	YES	3	512	200
M10	16	YES	10	512	40
M11	8	YES	9	128	100
M12	4	YES	5	256	102
M13	16	NO	3	512	80
M14	32	YES	10	1024	100
M15	8	YES	10	256	20
M16	32	YES	10	1024	20
M17	16	NO	16	512	17
M18	32	YES	15	1024	215
M19	16	YES	13	512	200
M20	16	YES	2	512	40
M21	8	YES	9	128	100
M22	4	YES	5	256	102
M23	16	NO	3	512	80
M24	32	YES	10	1024	300

Generate complex graphics with ggplot2

Seen as the new version of S, learn how to create truly impressive plots using R and the ggplot2 package

Advisor

Mihalis Tsoukalos is a UNIX administrator, a programmer (UNIX and iOS), a DBA and a mathematician. He has been using Linux since 1993

Resources

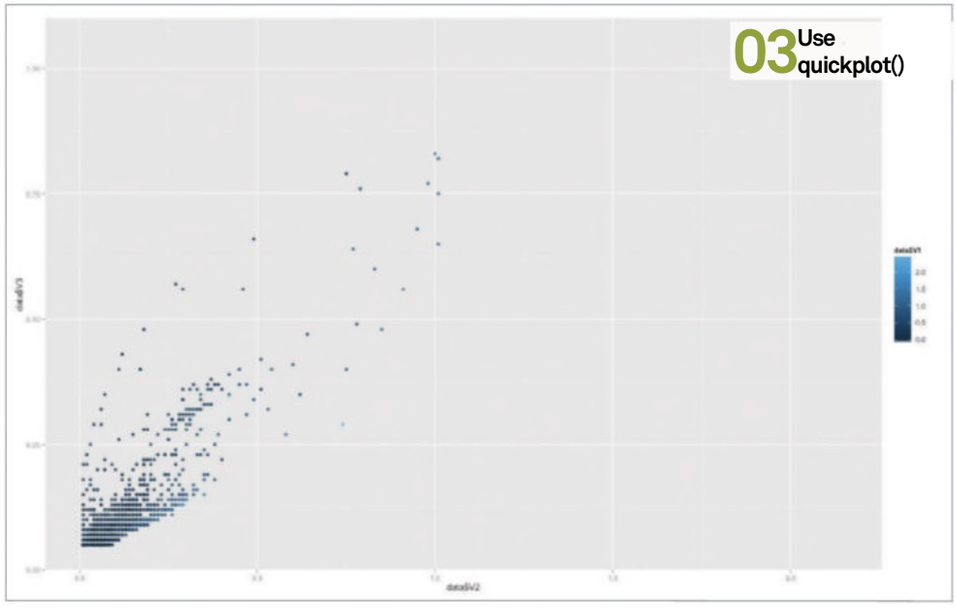
R Project r-project.org
RStudio rstudio.com
ggplot2 ggplot2.org
Documentation docs.ggplot2.org/current
RSQLite bit.ly/1ArJvkc

R is a GNU project based on S, which is a statistics-specific language and environment developed at the famous AT&T Bell Labs. You can think of R as the free version of S. Despite its simple name, R is a powerful piece of software for statistical computing with many capabilities and an interpreted programming language.

R packages can greatly extend its capabilities. Ggplot2 is an R package, written by Hadley Wickham, that is used for producing statistical and data graphics, working with plots in layers. Despite being a powerful package, it is reasonably easy to learn and produces sophisticated and beautiful plots that are of publication quality. Its main difference from most other graphics packages is that it has a deep principal grammar. Learning its grammar, which is based on the book The Grammar Of Graphics, will help you design better plots but it is not required in order to follow this article. In other words, the grammar tells that a plot is a mapping from data to aesthetic properties of geometric objects.

This article is full of practical examples that demonstrate the use of ggplot2 for drawing many different kinds of plots, including a plot of the Chrome's history file!

Feeling comfortable with mathematics and statistics is helpful but not vital for understanding this article.



"Experiment with your data and the various types of plots that R and ggplot can generate"

01 Install ggplot2

First run R. The ggplot2 R package isn't installed by default, so check if you already have it installed by running:

```
> require(ggplot2)
Loading required package: ggplot2
```

If it's not installed, download and select:

```
> install.packages("ggplot2")
```

If you execute the library() function without arguments, you'll get a list of installed packages. To get a detailed output, run the **installed.packages()** command without any arguments.

02 About R and data visualisation

R is a command line application, which is fine for plain text output but not for graphical output. RStudio is a more preferable graphical wrapper for R.

When visualising data, remember that not every plot suits every data set. This knowledge comes from experience, and experience comes from experimentation, so don't forget to experiment with your data and the various types of plots that R and ggplot2 can generate.

03 Use quickplot()

The ggplot2 package offers two main functions: quickplot() and ggplot(). The quickplot() function, qplot(), is similar to the plot() R function and is good for simple plots. The quickplot() function hides what happens underneath, whereas ggplot() is harder to use but is more flexible.

The following commands draw a plot using columns V2 and V3 from the data variable:

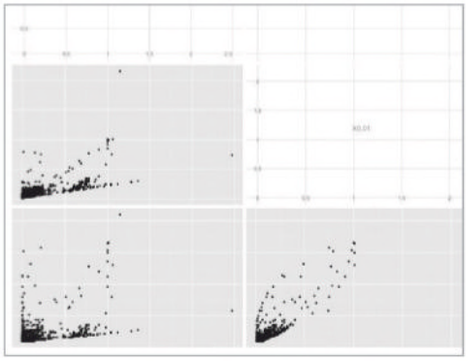
```
> str(data)
'data.frame':  16180 obs. of  3 variables:
 $ V1: num  0 0 0 0 0 0 0.98 1 1.06 1 ...
 $ V2: num  0.01 0.01 0.01 0.01 0.03 0.01
  0.58 0.85 1.01 1.01 ...
 $ V3: num  0.05 0.05 0.05 0.05 0.05 0.05
 0.27 0.48 0.65 0.75 ...
> quickplot(data$V2, data$V3)
```

The following version adds colour to the output:

```
> quickplot(data$V2, data$V3, color=data$V1)
```

04 Work with ggpairs()

The ggpairs() command finds relations between variables and then calculates the coefficient of correlation value. The coefficient of correlation is linked to the statistical correlation, a technique that shows whether or not two variables are related. As the coefficient of correlation approaches zero there is less of a relationship (no correlation), whereas the closer the coefficient is to -1 or +1, the stronger the correlation (positive or negative) is. A positive correlation shows that if one variable gets bigger then the other does as well. Conversely, a negative correlation denotes that if one variable gets bigger then the other becomes smaller.



The presented plot was produced using the following commands:

```
> data <-read.table("uptime.data",
header=TRUE)
> require(ggplot2)
> require(GGally)
> require(CCA)
> ggpairs(data)
```

05 Generate bar plots

Now use a sample dataset for plotting. The LUD dataset, available from **FileSilo**, is stored in a plain text file, named Lud.data. The titles of the columns are named to refer to their values. You can load the dataset into R using the following command:

```
> LUD <- read.table("lud.data",
header=TRUE)
```

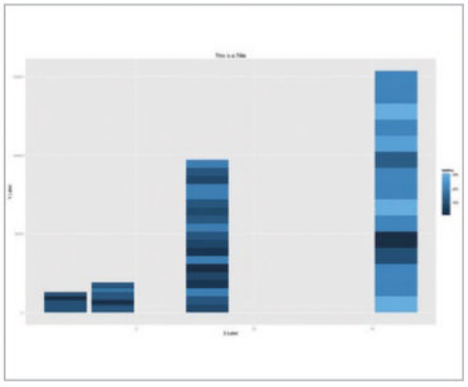
The bar plot is simple. This command generates it:

```
> ggplot(LUD, aes(x=RAM, y=SSD)) + geom_
bar(stat="identity")
```

If you type **ggplot(LUD, aes(x=RAM, y=SSD))** without specifying a plot, the command will show the 'Error: No layers in plot' message.

To change the colour of the bars, try the following variation:

```
> ggplot(LUD, aes(x=RAM, y=SSD,
fill=Uptime)) + geom_bar(stat="identity")
```



06 Add titles and labels

Sooner or later, it's likely that you will want to add a title and labels to the output. Adding a main title is simple – you just need to make use of the labs() function in order to do so. The previously plotted bar plot can thus be modified with inclusion of the the following command at the end:

```
> ggplot(LUD, aes(x=RAM, y=SSD, fill=Uptime)) + geom_bar(stat="identity") + labs(title="This is a Title")
```

Adding X and Y labels can be done by entering the following:

```
> ggplot(LUD, aes(x=RAM, y=SSD, fill=Uptime)) + geom_bar(stat="identity") + labs(title="This is a Title") + xlab("X Label") + ylab("Y Label")
```

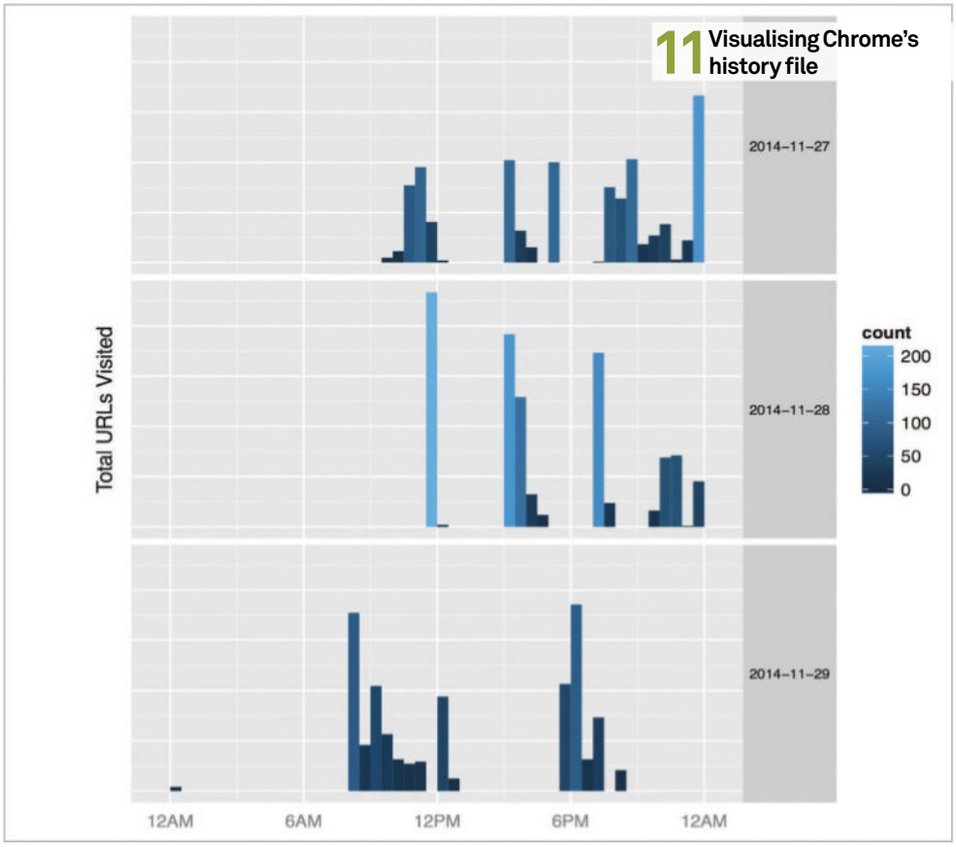
07 More about titles and labels

As well as add them, you can also change the appearance, size, font and colour of all the titles and labels. The following command makes the title blue and its size larger using the theme() function:

```
> ggplot(LUD, aes(x=RAM, y=SSD, fill=Uptime)) + geom_bar(stat="identity") + labs(title="This is a Title") + xlab("X Label") + ylab("Y Label") + theme(plot.title = element_text(size = rel(2), colour = "blue"))
```

To change the attributes of the X and Y axes, use the axis.line function:

```
> ggplot(LUD, aes(x=RAM, y=SSD, fill=Uptime)) + geom_bar(stat="identity") + labs(title="This is a Title") + xlab("X Label") + ylab("Y Label") + theme(plot.title = element_text(size = rel(2), colour = "blue"), axis.line = element_line(size = 3, colour = "red", linetype = "dotted"))
```



08 Create histograms

Generate histograms using the geom_histogram() function, similar to the geom_bar() function, and change the number of bars using the binwidth argument. Plot a simple histogram using the following command:

```
> ggplot(LUD, aes(Years)) + geom_histogram(binwidth=1, color='gray')
```

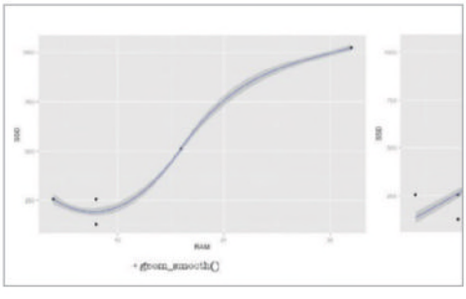
Using the geom_density() function you can draw a density plot:

```
> ggplot(LUD, aes(Years)) + geom_density(binwidth=1)
```

The following command draws a histogram and a density plot on the same plot:

```
> ggplot(LUD) + geom_histogram(aes(Years, ..density..), binwidth=2, color='white') + geom_density(aes(Years, ..density..), binwidth=2, color='red')
```

If you put the geom_density() command first, the histogram will be on top of the density plot and therefore the density plot will not be all visible.

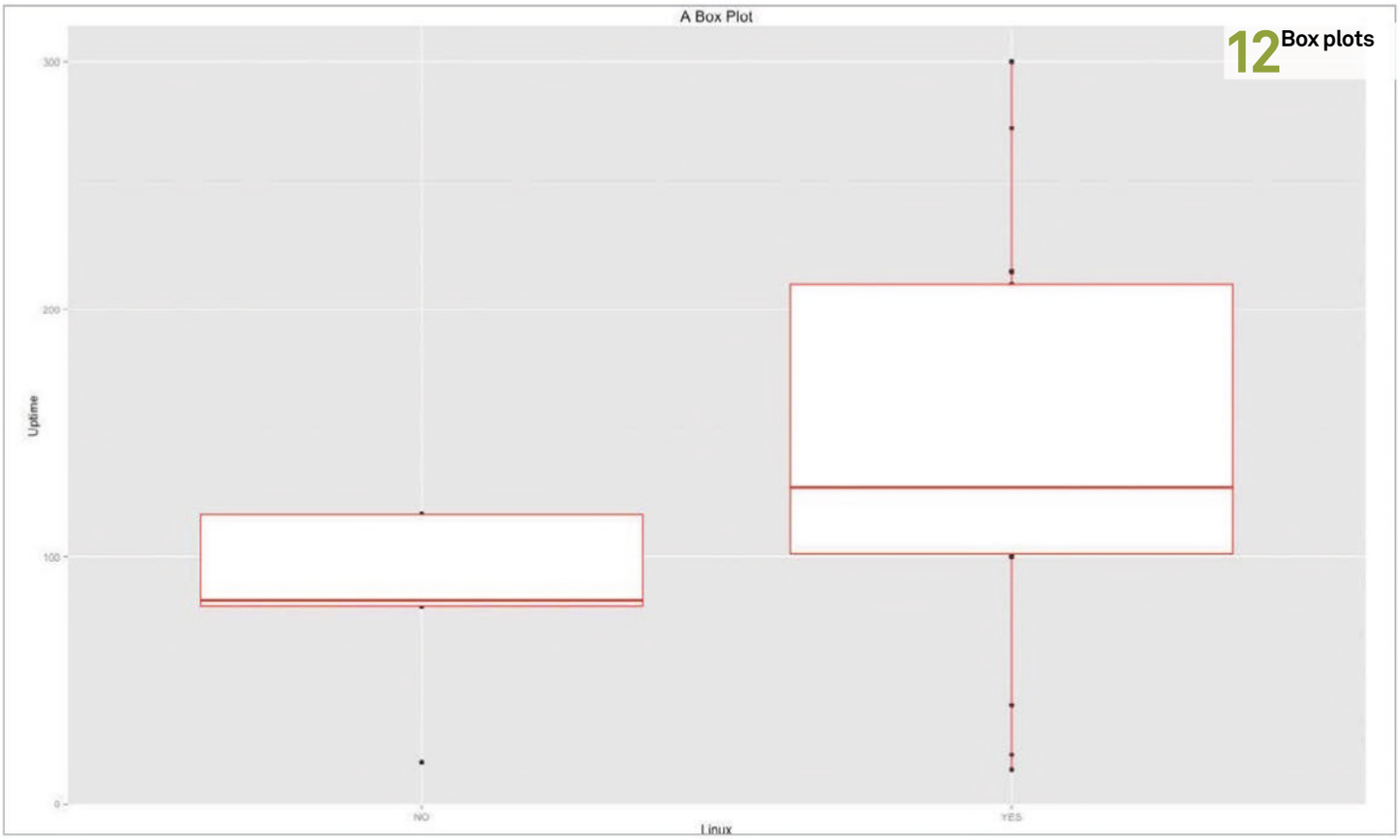


09 Add smooth layers

Another type of layer is the smooth layer. It doesn't display raw data, but rather a statistical transformation of the data.

The (method="lm") parameter generates a linear regression line instead of a LOESS (local polynomial regression fitting) curve, which is the default for samples with less than 1000 observations. For bigger samples, the default method is called GAM (generalised additive model). The produced plot was generated with the following commands:

```
> q <- ggplot(LUD, aes(x=RAM, y=SSD))
> q + geom_point() + geom_smooth()
> q + geom_point() + geom_smooth(method='lm')
```



12 Box plots

10 Work with shapes and facets

The following plot draws points using different shapes depending on the value of the non-continuous Linux variable:

```
> ggplot(LUD, aes(x=RAM, y=Uptime)) +
geom_point(aes(shape = Linux))
```

A facet allows you to split up your data by one or more variables and then plot the subsets of data together. Using facets is also a great way of generating conditional plots. Try the following point plot, which will generate two plots depending on the two different values of the Linux variable:

```
> ggplot(LUD, aes(x=RAM, y=Uptime)) +
geom_point() + facet_grid(Linux ~ .)
```

The facet_grid() function works fine when using continuous variables.

11 Visualising Chrome's history file

The history file of Chrome (simply called History) stores its history of visited websites in SQLite3 database format. Therefore, you can use the RSQLite R package to read it. The 'chrome.R' script generates an impressive output using RSQLite and ggplot2 with many layers. It can be done as follows:

```
$ ./chrome.R
Loading required package: methods
Loading required package: DBI
$ ls -l Rplots.pdf
-rw-r--r--@ 1 mtsouk  staff  5089 Nov 27
09:42 Rplots.pdf
```

The produced result is automatically stored in a file called 'Rplots.pdf' file.

12 Use box plots

A box plot can give you information regarding the shape, the variability and the median of a data set, quickly and efficiently. The presented box plot was generated using the following R command:

```
> ggplot(LUD, aes(Linux, Uptime)) + geom_
point() + geom_boxplot(colour = "red") +
labs(title="A Box Plot")
```

Based on the two discrete values of the variable, the output is divided into two subsets. For each subset, a separate box plot is produced individually.

13 Create R Scripts

It is very useful to learn how to create R scripts in order to use ggplot2 inside bigger scripts that can run as cron jobs. A sample script file, named 'ggplot.R' shows you how:

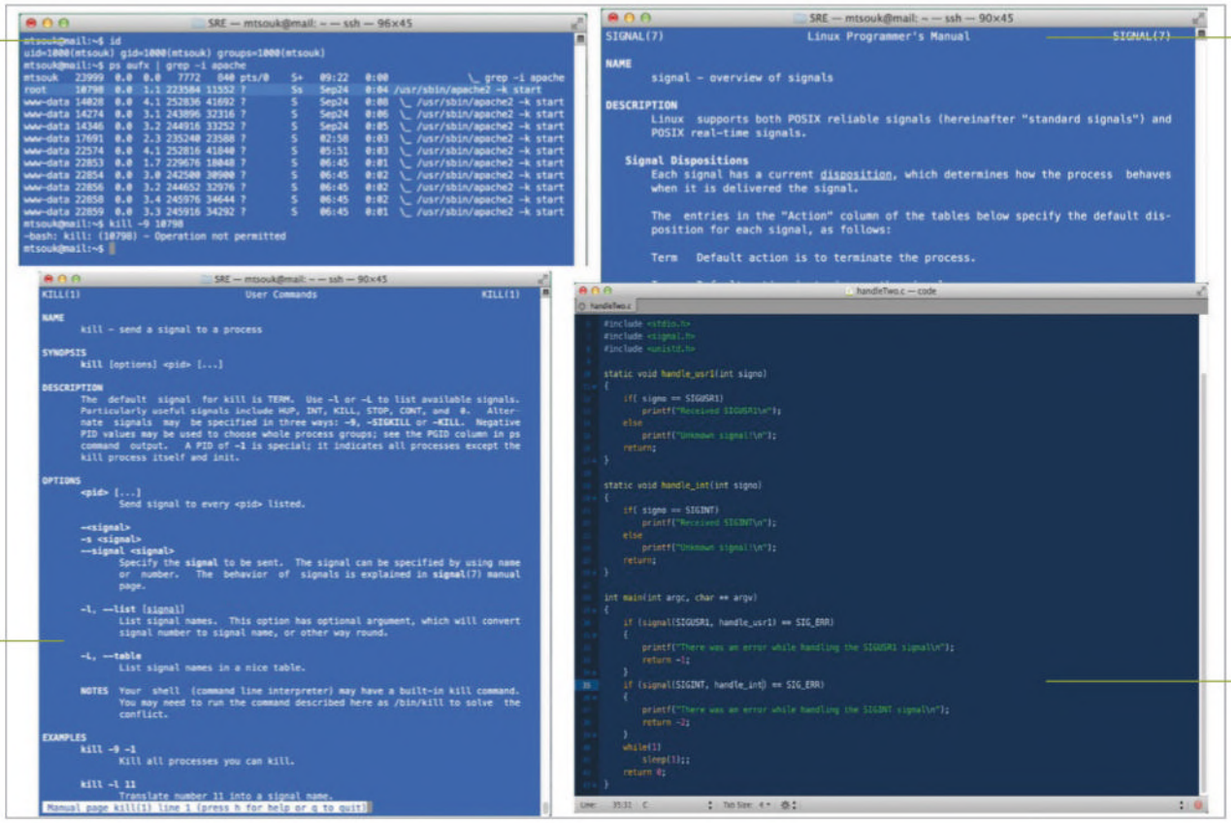
```
$ chmod 755 ggplot.R
$ ./ggplot.R
$ ll
total 160
-rwxr-xr-x@ 1 mtsouk  staff    234 Nov
14 22:41 ggplot.R
-rw-r--r--  1 mtsouk  staff  73820 Nov
14 22:43 ggplot.png
$ file ggplot.png
ggplot.png: PNG image data, 1280 x 800,
8-bit/color RGBA, non-interlaced
```

14 Final thoughts

The more you use ggplot2 and the better you know your data, the better the output you'll achieve. Just don't forget that ggplot2 works using layers! Also, to take full advantage of plotting you'll have to plot the right metrics, and finding the right metrics is not always simple, but once you have the hang of it, it will become second nature.

Sending a signal to a process without having the necessary permissions is not allowed

A small part of the man page of the signal system function: man 7 signal



A part of the man page of the kill command which can be seen by executing “man kill”

This is the full C code for the basic handling of the SIGINT and SIGUSR1 signals

Master UNIX signal handling

Learn how to utilise signals and program their handling in C

Advisor

Mihalis Tsoukalos
is a UNIX administrator, a programmer (UNIX and iOS), a DBA and a mathematician. He has been using Linux since 1993

Resources

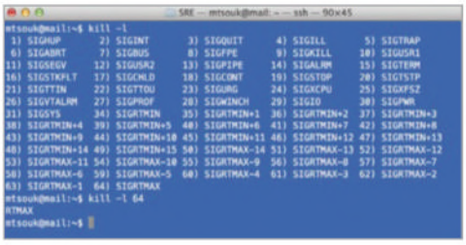
A text editor
A C compiler

UNIX signals are software interrupts that offer a way of handling asynchronous events on a UNIX system. Every application, apart from the trivial ones, must be able to deal with signals. This article will introduce you to the most important signals and show you ways of handling them in your Linux applications.

Each signal can be identified by name or numeric value, but using the signal name is easier to remember and the recommended way. In order to send a signal to a running application, you should have the required UNIX privileges. If you are the root user, you can send any running process any signal you want. Signals are important for avoiding blocking situations and blocking can happen when waiting for user input, reading a file or reading from a device.

Example code has been uploaded to FileSilo.co.uk/bks-835 for you to follow along with. The examples use the printf() C function inside signal handler functions for educational reasons only; it is considered very bad practice to use them in production code as it can introduce nasty bugs.

Signals can also be handled in other programming languages, including Perl and Python, as you will see at the end of the article.



01 About signals

All signal names begin with SIG, which is sometimes not mentioned in the documentation available online. The following command shows a list of all signals:

```
$ kill -l
```

All signals have a default action but most of them allow you to bypass the default action by writing your own code.

```
mtsouk@mail:~$ kill -l TERM
15
mtsouk@mail:~$ kill -l 15
TERM
mtsouk@mail:~$
```

02 The kill command

The kill command sends the TERM signal by default. Therefore the following two commands are equivalent:

```
$ kill -TERM <process_id>
$ kill <process_id>
```

You should avoid sending the SIGKILL signal at a running process unless it is absolutely necessary. The KILL signal is noncatchable and nonignorable signal that violently terminates a process without allowing it to clean up properly.

If you run the kill command followed by the –l option followed by a number it will return the signal name. Similarly, if you give a signal name, it will return the signal number.

03 The three most common UNIX signals

Have you ever pressed Ctrl+C in order to stop a program from running? Ctrl+C sends the SIGINT signal to the program; SIGINT is the most commonly used signal.

Another important signal is SIGKILL, which happens when you kill a process using the kill -9 command. As you will see in the next step, SIGKILL is a special kind of signal.

Another useful signal is called SIGHUP, which is commonly used to notify server processes to reread their configuration files.

handleOne.c

06 Explain the C code

```c
#include <stdio.h>
#include <signal.h>

static void handle_usr1(int signo)
{
    if( signo == SIGUSR1)
    {
        printf("Received SIGUSR1\n");
    }
    else // This should not happen
    {
        printf("Unknown signal!\n");
    }
    return;
}

int main(int argc, char ** argv)
{
    if( signal(SIGUSR1, handle_usr1) == SIG_ERR)
    {
        printf("There was an error while handling the SIGUSR1 signal\n");
        return -1;
    }

    while(1)
        ;

    return 0;
}
```

Line: 13:3 | C | Tab Size: 4 | handle_usr1

"The KILL signal is similar to unplugging from the mains"

04 Signal handling

A program cannot handle all signals; some of them are noncatchable and nonignorable. The signals SIGKILL and SIGSTOP cannot be caught, blocked or ignored. The reason for this is that they provide the kernel and the root user a way of stopping any process.

The number for the KILL signal is 9. It is usually called in extreme conditions where you need to act fast, so it is the only signal that is usually called by number because it is quicker to type one number than to type in a phrase.

The effect of the KILL signal is similar to unplugging your computer from the mains instead of powering it down normally and can cause various problems, especially when it is used to stop server processes such as database, web and email servers.

05 Handling just one signal

Our example C program 'handleOne' handles the SIGUSR1 signal. SIGUSR1 is a user-defined signal for use in application programs and its default action is 'terminate'. By handling it, you change its default action.

Apart from the USR1 signal that is specifically handled, all of the other signals will be handled using their default behaviour. You can use Ctrl+C to stop the program or the kill command from another terminal.

06 Explain the C code

The C code is easy to understand. The signal function defines that the handle_usr1 function will be called when a SIGUSR1 signal occurs. You can refer to a signal either by name or by number inside your code, but it is advised to refer to it by its name rather than by its signal to make your code easier to read.

The endless while() loop is used for making the program run forever in order to be able to test signal handling and, usually, server processes will have a similar code.

07 The signal() function

The signal() function, which we use in our example code, takes two arguments. The first argument is the name or the number of the signal. The second argument is a function name: the signal handler. If the value is SIG_DFL, default handling for that signal will successfully occur. However, if the value is SIG_IGN then the signal will be ignored. Otherwise, it must be a valid function name that will be called when that signal occurs –in the case of our example, this function is handle_usr1. New applications should use sigaction() instead of signal().

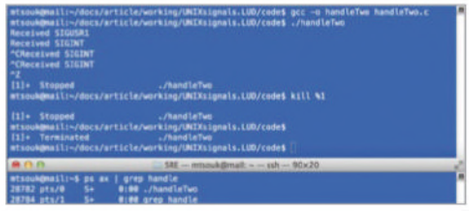
```
// Programmer: Mihalis Tsoukalos
// Date: Thursday 25 September 2014
//
// Using the SIGALRM signal

#include <stdio.h>
#include <unistd.h>
#include <signal.h>

void handle_sigalrm(int);

int  main(int argc, char **argv)
{
    char name[100];

    signal(SIGALRM, handle_sigalrm);
    alarm(10);

    while (1)
    {
        printf("You only have 10 seconds to type your name
        scanf("%s", name);
        break;
    }

    printf("Hello %s\n", name);

    return 0;
}

void handle_sigalrm(int signo)
{
    printf("Too late... ");
    printf("Please try again...\n");
    alarm(10);
}
```

08 The SIGALRM signal

Imagine that you have a program that waits for user input but, for some reason, the user does not type anything or does not know that he or she has to type. Should the computer wait forever? The solution to this kind of problems is given by the SIGALRM signal. The alarm() system call is really a timer that allows you to receive SIGALRM in a preconfigured number of seconds (see 'signalALARM.c').

Compiling and running the code produces the following output:

```
$ gcc -Wall -o signalALARM signalALARM.c
code$ ./signalALARM
You only have 10 seconds to type your
name: Too late... Please try again...
Too late... Please try again...
Mihalis
Hello Mihalis
```



```
mtsouk@mail:~/docs/article/working/UNIXsignals.LUD/code$ gcc -o handleTwo handleTwo.c
mtsouk@mail:~/docs/article/working/UNIXsignals.LUD/code$ ./handleTwo
Received SIGUSR1
Received SIGINT
^CReceived SIGINT
^CReceived SIGINT
^Z
[1]+  Stopped                 ./handleTwo
mtsouk@mail:~/docs/article/working/UNIXsignals.LUD/code$ kill %1

[1]+  Stopped                 ./handleTwo
mtsouk@mail:~/docs/article/working/UNIXsignals.LUD/code$
[1]+  Terminated              ./handleTwo
mtsouk@mail:~/docs/article/working/UNIXsignals.LUD/code$
```



```
mtsouk@mail:~$ ps ax | grep handle
28782 pts/0    S+     0:00 ./handleTwo
28784 pts/1    S+     0:00 grep handle
```

09 Handling two signals

The code of our example 'handleTwo.c' program is in many ways similar to the handleOne.c code. This time, your program will handle two signals, SIGUSR1 and SIGINT, using a different function for each signal. Separating the two handling functions makes your life easier but you will then need to write more code.

The problem is that you have to explicitly add the signals you want to support, which is especially tedious if you are planning on supporting more signals, but there is no other way to do it. You will learn how to block multiple signals in Step 14.

```
// Using the raise() function call

#include <stdio.h>
#include <signal.h>
#include <unistd.h>

static void handle_usr1(int signo)
{
    if( signo == SIGUSR1)
    {
        printf("Received SIGUSR1\n");
    }
    else // This should not happen
    {
        printf("Unknown signal!\n");
    }
    return;
}

int main(int argc, char ** argv)
{
    if( signal(SIGUSR1, handle_usr1) == SIG_ERR)
    {
        printf("There was an error while handling the SIGUSR1 signal\n");
        return -1;
    }

    while(1)
    {
        raise(SIGUSR1);
        sleep(30);
    }

    return 0;
}
```

10 The raise() function

A process can send itself a signal with the help of the raise function. Its very simple to use and you should have no problem including it in your programs. Unlike the signal function, which takes two arguments, the raise function accepts just one argument: the name or the number of the signal (see 'raise.c').

11 Real work with signal handling

The single most important task of a signal handler function is to make sure that the signal does not do any damage to data. Think of it as an object destructor in object-oriented terminology. The most common job of a signal handler function is to gracefully close files or connections, write data to disk and then allow a program to exit.

12 How Apache handles signals

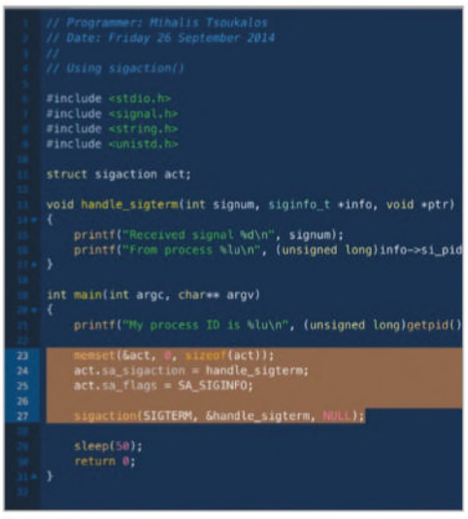
The Apache parent process can handle the TERM, USR1, HUP and WINCH signals. The TERM signal ends the Apache process in a bad way and should therefore only be used in extreme situations.

The USR1 signal causes the Apache parent process to advise the children to exit after serving their current request. After all the children are done, the parent rereads its configuration files, reopens its log files and restarts then child processes. If the new Apache configuration file has errors in it, then Apache will not restart: it will exit with an error.

The HUP signal does the same job as the USR1 signal but with a big difference. Instead of waiting for the children to exit gracefully, it just kills the children.

The WINCH or graceful-stop signal causes the Apache parent process to advise the children to exit after serving their current request, remove its PID file and stop listening to ports without quitting. After the termination of all children, it will also quit itself. This functionality is very helpful when you are upgrading Apache but it can cause deadlocks and race conditions sometimes.

On a Debian 7 system, the process id of the Apache parent process can be found in the /var/run/apache2.pid file.



```
// Programmer: Mihalis Tsoukalos
// Date: Friday 26 September 2014
//
// Using sigaction()

#include <stdio.h>
#include <signal.h>
#include <string.h>
#include <unistd.h>

struct sigaction act;

void handle_sigterm(int signum, siginfo_t *info, void *ptr)
{
    printf("Received signal %d\n", signum);
    printf("From process %lu\n", (unsigned long)info->si_pid
}

int main(int argc, char** argv)
{
    printf("My process ID is %lu\n", (unsigned long)getpid()

    memset(&act, 0, sizeof(act));
    act.sa_sigaction = handle_sigterm;
    act.sa_flags = SA_SIGINFO;

    sigaction(SIGTERM, &handle_sigterm, NULL);

    sleep(50);
    return 0;
}
```

13 The sigaction() system call

The sigaction() system call has the same basic effect as signal(). Sigaction can offer more control but it also adds more complexity. In particular, sigaction can let you specify additional flags to control when the signal is generated and how the handler is invoked. Once a signal handler is installed, it normally remains installed until another sigaction() system call is made.

Its first argument specifies a signal number as usual. Both the second and third arguments are pointers to a structure called sigaction. This structure specifies how the process should handle the given signal. (See 'sigaction.c'.)

It is highly recommended that you use sigaction() instead of signal() because sigaction is more reliable.

14 The sigsuspend() and the sigprocmask() functions

The combination of these two function calls allows you to block and unblock selected signals. They are usually used for protecting critical regions of code from being interrupted by other signals. The collection of signals that are currently blocked is called the signal mask. Each process has its own signal mask.

First, you block the signals with sigprocmask and then after the critical code has completed, you run sigsuspend with the signal mask that was returned by sigprocmask – it is a useful yet advanced technique.

As you already know from Step 04, it is impossible to block SIGKILL or SIGSTOP even if you specifically add them into a signal mask.

15 Signal sets

A very important feature is the ability to include multiple signals in signal sets. You will need the sigset_t data type to represent a signal set and five functions to manipulate them: sigemptyset() (clears the mask), sigfillset() (sets all bits in the mask), sigaddset() (sets the bit that represents a certain signal), sigdelset() (clears the bit that represents a specific signal) and sigismember() (checks the status of a certain signal in a mask). The presented example ('signalSet.c') handles SIGUSR1 and SIGINT and blocks all other signals.

```
#!/usr/bin/perl -w

use warnings;
use strict;

$SIG{INT} = sub { print "Caught a SIGINT Signal: $!\n" };
use sigtrap qw/handler error_signal_handler error-signals/;

sub error_signal_handler
{
    print "An error signal caught!\n";
}

while(1)
{
    sleep(30);
}
```

17 Signal handling in Python

"Sigaction offers more control but it also adds complexity"

16 Signal handling in Perl

Perl has two ways of handling signals: using the %SIG hash or using the sigtrap pragma. The sigtrap pragma understands three groups of signals: normal-signals (HUP, PIPE, INT and TERM), error-signals (ABRT, BUS, EMT, FPE, ILL, QUIT, SEGV, SYS and TRAP) and old-interface-signals (ABRT, BUS, EMT, FPE, ILL, PIPE, QUIT, SEGV, SYS, TERM and TRAP).

Our example ('signals.pl') uses both methods, but usually you only use one of them.

```
$ ./signals.pl
^CCaught a SIGINT Signal: Interrupted
system call
^Z
[1]+  Stopped    ./signals.pl
$ bg
[1]+ ./signals.pl &
$ ps ax | grep signals | grep -v grep
32431 pts/0    S      0:00 /usr/bin/perl
-w ./signals.pl
$ kill -ABRT 32431
$ An error signal caught!
$ kill -BUS 32431
$ An error signal caught!
$ kill -HUP 32431
$
[1]+  Hangup     ./signals.pl
$ ps ax | grep signals | grep -v grep
```

```
import os, sys
import signal
import time

def sigint_handler(signum, frame):
    print 'Stop Interrupting me!'

signal.signal(signal.SIGINT, sigint_handler)

def main():
    while True:
        print '#'
        time.sleep(5)

if __name__ == "__main__":
    main()
```

17 Signal handling in Python

The Python handling of signals is similar to Perl. The example Python program ('signals.py') handles the SIGINT signal because it is the one most frequently used.

```
$ python signals.py
#
^CStop Interrupting me!
#
^CStop Interrupting me!
#
^Z
[1]+  Stopped    python signals.py
$ kill %1
[1]+  Stopped    python signals.py
[1]+  Terminatedpython signals.py
```

Build a RAID array

Use RAID to create faster and more secure storage systems in your PC or server

Advisor

Rob Zwetsloot
models complex systems and is a web developer proficient in Python, Django and PHP. He loves to experiment with computing

Resources

mdadm

neil.brown.name/blog/mdadm

Hardware RAID support

It appears that we're currently seeing a plateau on storage in hard drives. While we're up to about 6 TB in storage on drive, 4 TB drives have been in circulation for years now. Luckily, the jump from 4 to 6 TB is larger than the doubling in size of old – one terabyte is still a lot of data, after all – but as always, files are getting larger and larger, and people are amassing more and more data.

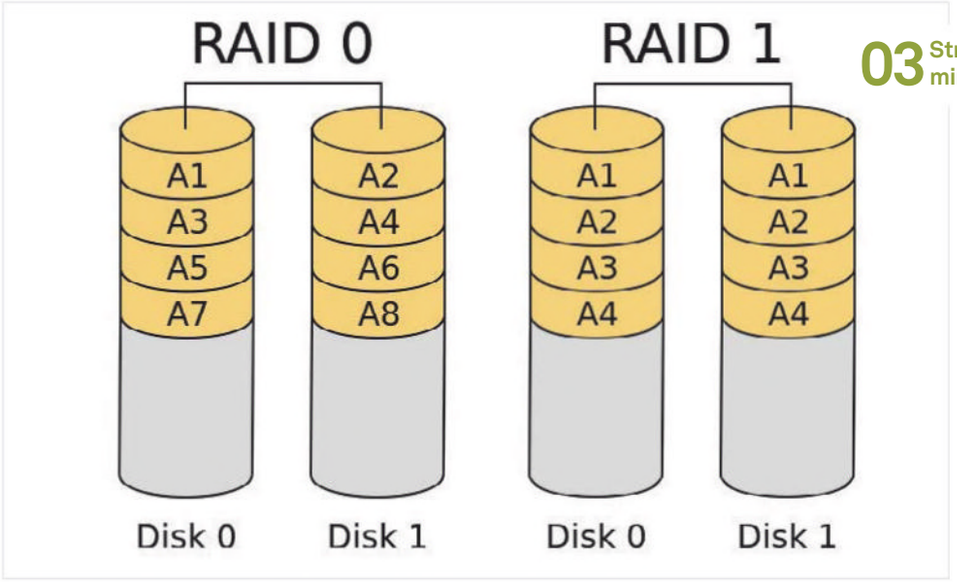
There have been methods in existence for a long time to pool multiple hard drive resources in order to create larger storage solutions – the benefits of which normally involve increased read and write speeds as well as large storage sections overall.

In this tutorial we'll teach you everything you need to know to set up your own RAID array. This involves not only doing it via hardware RAID, but also a software RAID in Linux. We'll also talk about what you need to know when actually selecting hard drives, which RAID levels you should be looking at and what the benefits of them are.

01 Hard drive selection

For all the different versions of RAID we'll be using, at the very least you need to use hard drives with the exact same storage – so if you had one 2 TB hard drive, they'd need to all be 2 TB. This has a lot to do with the way the RAID levels write data, requiring each read and write operation to be replicated over each drive as part of the system. This means hard drives must be exactly same size for the files to be written properly.

While you can do it with any hard drives of the same size, it's far more recommended to get the



Striping, to the left, splits the data across drives, whereas mirroring duplicates it

03 Striping and mirroring

"Methods to pool multiple hard drive resources to create larger storage solutions have existed for a long time"

same make and model of hard drive to get the best experience out of the RAID. There's even an argument for getting similar batches of hard drives for the task. Not only does this allow for even greater parity between the hard drives, it also makes it easier to replace any broken drives in the array. As for write speed, as long as all the drives are the same speed (which will be fine with the same model of drives), there are no problems with using 5400, 7200 or any other speed of hard drive you have.

02 Choosing a RAID type

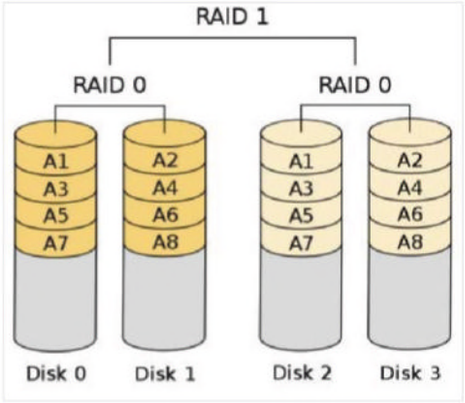
RAID comes in different types, each numbered and starting at 0. The higher then number, the more complex the system will be (apart from RAID 10, which is a common abbreviation of RAID 1+0, which we'll cover in a moment). Choosing a RAID type depends entirely on your needs, budget and the importance of preserving the data that will be stored on the drives.

For most home uses, the RAID 0, RAID 1 and RAID 1+0 techniques may be the best. These allow you to increase read speed and in two cases create a larger storage area than any single drive. Even better, any failures can be easily fixed for two of them by just putting new hard drives into the array.

For more advanced use, in all likelihood business use, RAID 5 and RAID 6 allow for more storage to be used over multiple drives while still having the levels of redundancy and backup that you see in the lower levels.

03 Striping and mirroring

RAID 0 and RAID 1 are the most common types of RAID arrays, and both involve one of the two important methods of using RAID. RAID 0 arrays are also called striped volumes, whereas RAID 1 arrays are otherwise known as mirrored volumes.



At its core, RAID 0 enables you to join two hard drives together of the same size and use the storage of both to create one large 'hard drive' that the system can see. It does this by writing data over both drives for individual files and such, which is usually illustrated as stripes. The other benefit you get from RAID 0 is speed, however if one hard drive fails then you will lose all your data.

RAID 1 is the opposite of RAID 0: instead of doubling the size of one hard drive's storage by combining them, instead you create a dedicated backup for the hard drive with each piece of data written twice. Read speed increases as before, although write speed is the same as any one disk (the RAID array deals with duplicating the data once it's written).

The dangers

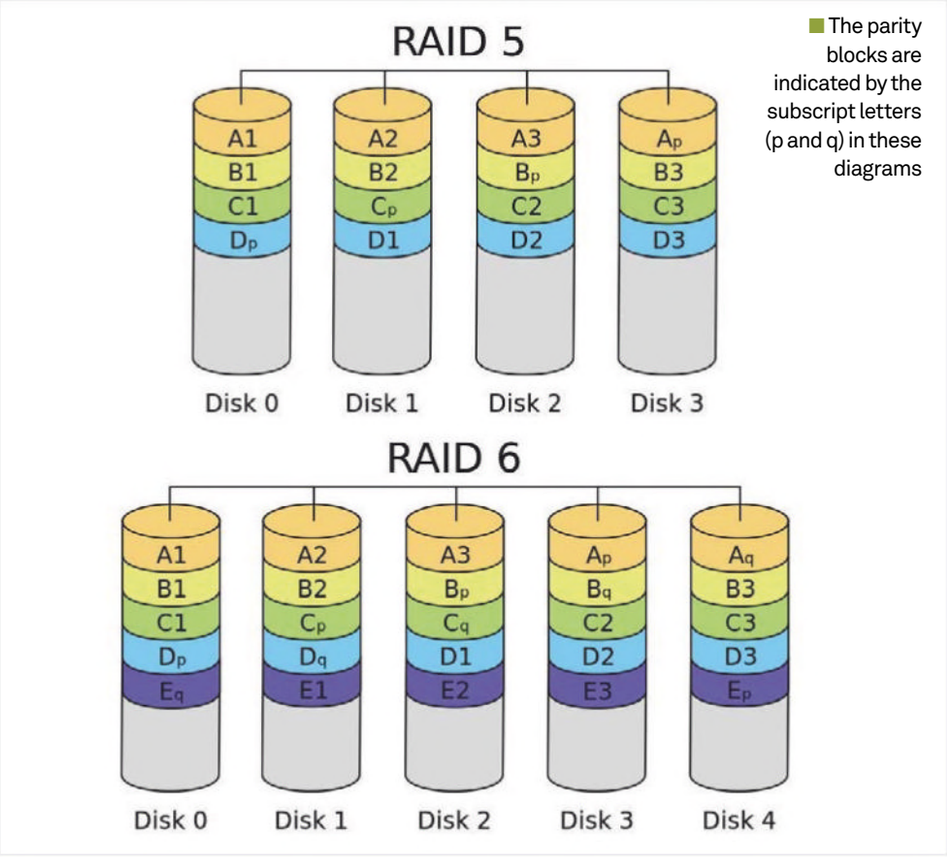
RAID 0 and 1 are the simplest RAID levels you can use, however any version of RAID can cause problems when trying to move hard drives or upgrade in general. The RAIDs are very dependent on each other, and it can fall apart if you don't keep it maintained very well.

04 Powers combined

RAID 1 and RAID 0 can be combined into RAID 1+0, which offers both mirroring and striping. It requires at least four hard drives to work, and always needs to be an even number of hard drives.

RAID 10 works by having pairs of hard drives that are setup as RAID 1 (mirroring), with each set then combined together with RAID 0 for striping. This nested RAID allows for both the read and write performance of RAID 0, the increase in storage space from the same, along with the redundancy and backups from the RAID 1 mirroring. If one hard drive dies, it can easily be replaced in order to get the RAID array going again.

This is the simplest way to both increase the storage capacity of one array, while also having a way of recovering from hard drive failure. However, it has the most inefficient use of the space compared to some of the higher RAID versions.



The parity blocks are indicated by the subscript letters (p and q) in these diagrams

Generally though, this setup can be quite good at home in desktops and even on home servers.

05 Block-level striping

The other two big RAID array types are RAID 5 and RAID 6, which use block striping and advanced calculations to create backups and parity across drives, ultimately allowing you to use more of the data space across your hard drives.

RAID 5 requires a minimum of three drives, with a maximum of 32 allowed in the array. What RAID 5 does is have one drive in every stripe of data include a parity block – this block acts as a sort of backup and allows for calculations of whatever is missing if one of the drives die. This parity block is distributed evenly across the different hard drives, which allows for quick read requests.

RAID 6 requires a minimum of four disks and introduces the concept of parity striping: parity blocks are split up over two drives for increased security against hard drive failures. This comes at the expense of the extra storage you'd get in a RAID 5 array, and write speed does not increase over normal drives.

06 Other RAIDs

There are other RAID levels, however these four and the combinations of them are the most common and useful out of them. RAID 2 and 3 can have excellent transfers rates but are optimised for one operation at a time, and cannot handle multiple requests very well – these kinds of arrays are recommended for video editors that require the reading and writing of large files often.

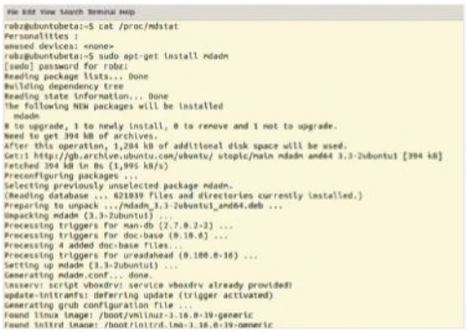
You can also do more combinations as you see fit, using RAID 0 to join multiple RAID 5 or RAID 6 arrays together to create more space while still having a level of parity on each nested array. These can be used to get around hard drive limits in massive servers or virtual systems, along with RAID 100 that has you create RAID 0s with RAID 1+0 setups. While arrays like this do have backups and fail-safes, their general complexity make it more difficult to maintain and repair in the long run.

07 Hardware RAIDs

A hardware RAID allows you to set up a RAID array on a hardware level, which then presents Linux with a single hard drive that it can use as normal. The RAID levels you can use depend on the RAID that your motherboard or RAID card support. Not all RAID cards have Linux drivers either, so you need to make sure you get the exact right card for the job.

Setting up a RAID in this way is usually dependent on what kind of card or BIOS you're using for it. You'll need to first install the drives while the system is off though, and then look for a RAID setup utility during the POST part of when you turn the system on. Refer to the user manual for your mobo or card to find the exact the steps to then set up each RAID type on the provided hard drives.

Once that's complete, all you'll need to do is format the drive in Linux to use it as normal.



08 Software RAID

Setting up a software RAID is supported by Linux at the kernel level, and can allow for more flexibility depending on the hardware on the system as it supports all the above RAID types we've discussed without requiring any specific hardware (other than the space and slots to add enough hard drives).

You'll need to prepare your system for adding the RAID by checking if it first has the right modules in the kernel – you can check that by calling:

```
# cat /proc/mdstat
```

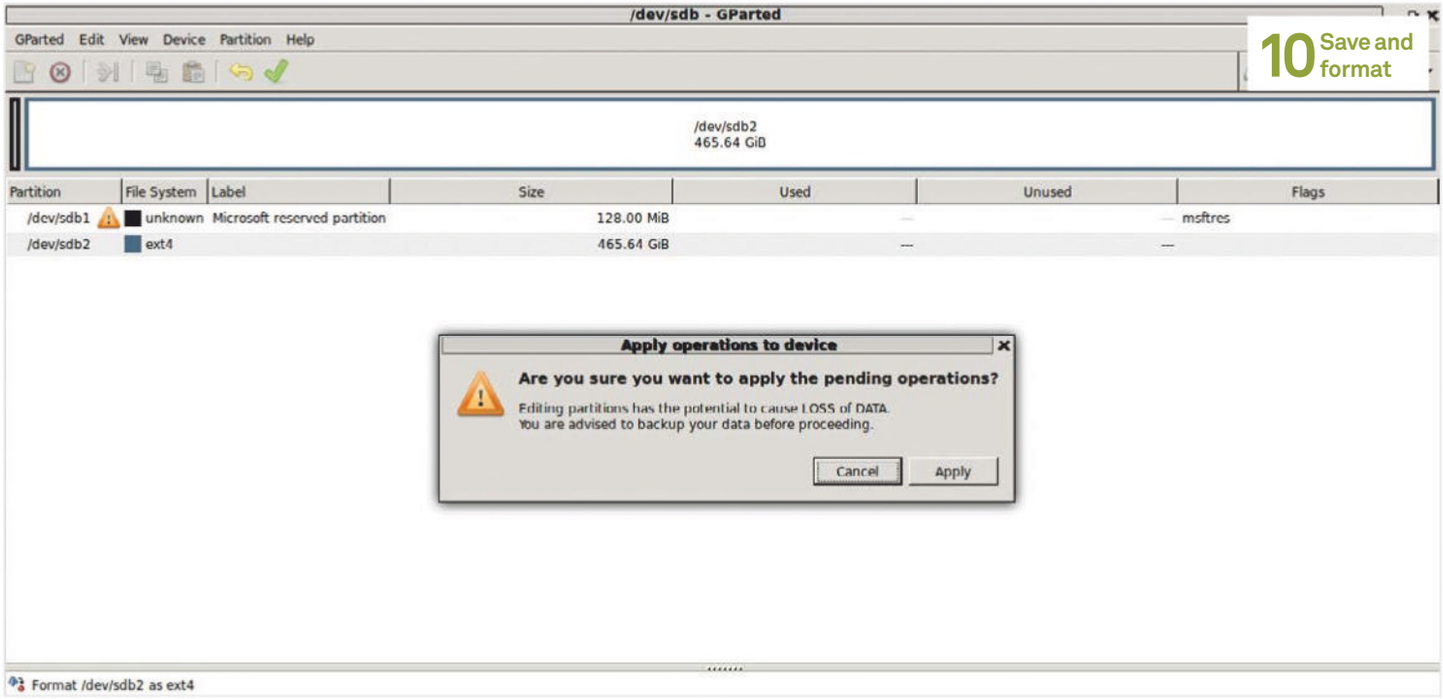
If the file exists then you likely have the right modules installed (although doing a **modprobe raid456** wouldn't hurt to make sure it's loaded properly). You'll also need to install mdadm, the multiple-disk administration tool:

```
# apt-get install mdadm
```

You can use mdadm to not only create the software RAIDs, but also to manage them, redo them, rebuild them and even more advanced RAID tasks

09 Setting up a software RAID

To set up a software RAID, you first need to make sure that all of your hard drives are installed into your system. For this example, we're assuming you already have Linux set up on the device on a completely different hard drive, but you can use similar settings to create the RAID in a live disc



before you install to it. Boot into Linux and open the terminal to begin.

First of all, we need to figure out what the hard drives are called on our system, so use **fdisk -l** to get a list of their addresses (/dev/sdb, /dev/sdc, etc). Note them down and we can set up out RAIDs.

```
RAID 0:

# mdadm --create --verbose /dev/md0
--level=stripe
--raid-devices=2 /dev/sdb1 /dev/sdc1

RAID 1:

# mdadm --create --verbose /dev/md0
--level=mirror
--raid-devices=2 /dev/sdb1 /dev/sdc1

RAID 10:

# mdadm -create --verbose /dev/md0
--level=10
--raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/
sdd1 /
dev/sde1

RAID 5:

# mdadm -create --verbose /dev/md0 --level=5
--raid-devices=3 /dev/sdb1 /dev/sdc1 /dev/
sdd1
```

"Setting up a software RAID is supported by Linux at the kernel level, and can allow for more flexibility"

```
RAID 6:

# mdadm -create --verbose /dev/md0 --level=6
--raid-devices=3 /dev/sdb1 /dev/sdc1 /dev/sdd1 /
dev/sde1
```

10 Save and format

Once you're happy with the array (or arrays) you have created, you need to then save it. You can do this on an Ubuntu server by using:

```
# mdadm --detail --scan >> /etc/mdadm/
mdadm.conf
```

Or on most other distros with:

```
# mdadm --detail --scan >> /etc/mdadm.
conf
```

Run **cat /proc/mdstat** again to see which arrays are now listed in the file, and it should have the one you've just set up. If it's there, all you now need to do is format to whatever filesystem you want using your command line or GUI tool of choice – as the kernel is handling the RAID communication, you won't have to worry about erasing any information. Add it to /etc/fstab so it mounts at boot, and your RAID is complete!

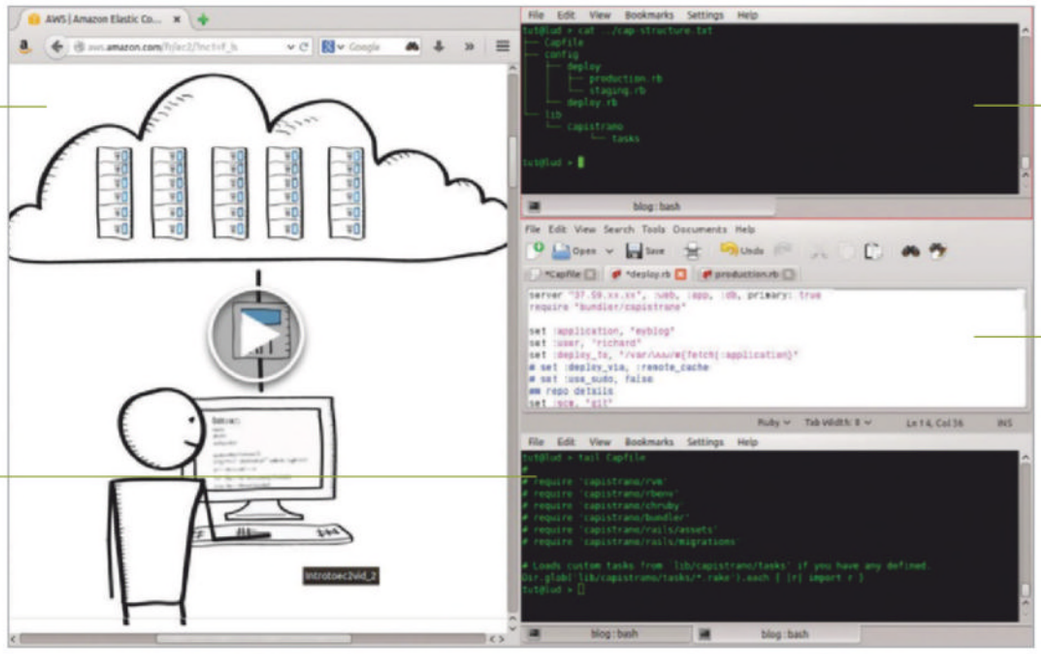
It's worth digging into the man page for mdadm to learn how you can use it to manage your RAID in the future, and keep it in top condition.

JBOD

A much simpler version of RAID is JBOD, or Just a Bunch Of Disks. These connect hard drives together to increase the storage capacity and split it across the drives. This way, if one drive dies then you only lose the files that are saved across them, but it will still work without needing a replacement.

No matter where you deploy your app, Capistrano makes it easy to deploy Rails (and other platforms) to *nix servers

Changes like adding a QA stage after production, with qa.rb, are both possible and easy with its modular structure



Built from Ruby, and originally for Rails, Capistrano 3 is a great fit for deploying other web platforms

Capistrano's flexibility and power can mean a lot of config, but it's relatively straightforward

Continuously deploy web apps with Capistrano

Advisor

Richard Smedley
A Unix jack-of-all-trades, Richard doesn't spend enough time in any language to get truly proficient, but always has a shell open so learnt scripting by osmosis

Resources

Ruby >= 1.9 (RBENV makes it easier)

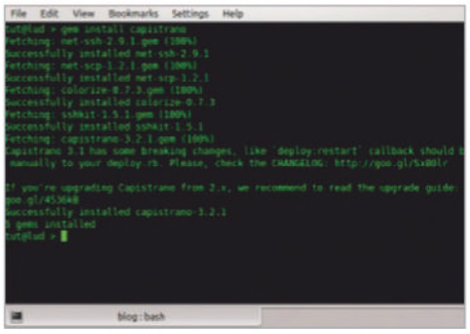
git www.git-scm.com

Capistrano 3 www.capistranorb.com

Move your web apps' development versions painlessly from staging and testing to deployment on databases and web servers

Capistrano automates deploying web apps to your servers, taking care of tiresome tasks like running a series of remote commands on any box on which you have SSH access. Capistrano's main recipe is deploy, containing tasks such as rollback, which consists of groups of commands. Servers are given roles: app – application servers, web – web servers, and db – database servers. Within this (expandable) framework, it's easy to adjust Capistrano's configuration files for your particular app and collection of servers. Basic set up is simple enough, but Capistrano 3 is a flexible framework and you will need to go a long way beyond this simple introduction if you want to begin to get the full use of it.

We will need a recent Ruby, Git (plus a GitHub account) and a Rails project you're working on. Also rbenv makes working with Rails easier, and capistrano-rbenv makes sure Capistrano uses the correct rbenv version of Ruby for deployment. It doesn't matter whether you run Passenger, Unicorn or Puma as your app server, but note that there are Capistrano plugins to help with extra tasks on all of them. Search **rubygems.org** for the full choice.

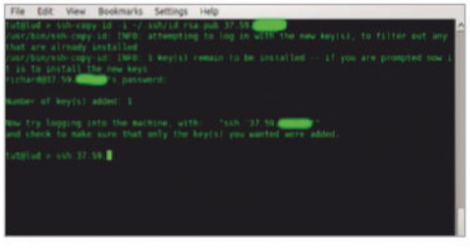


01 Install Capistrano

Essentially, Capistrano copies your app from Git, over SSH, to your server and takes care of all of the operations you'd have to do if you were to move the files by hand. Capistrano will deal with any database migration, changes to files and file names, restarting the web server and so on. Installing Capistrano, and the extras package which is nice to have, is a simple:

```
gem install capistrano
```

Any problems, check your Ruby installation. The capistrano-ext gem you may see referred to elsewhere is no longer essential, as features like extra staging options have been integrated into the main codebase.



02 SSH Keys

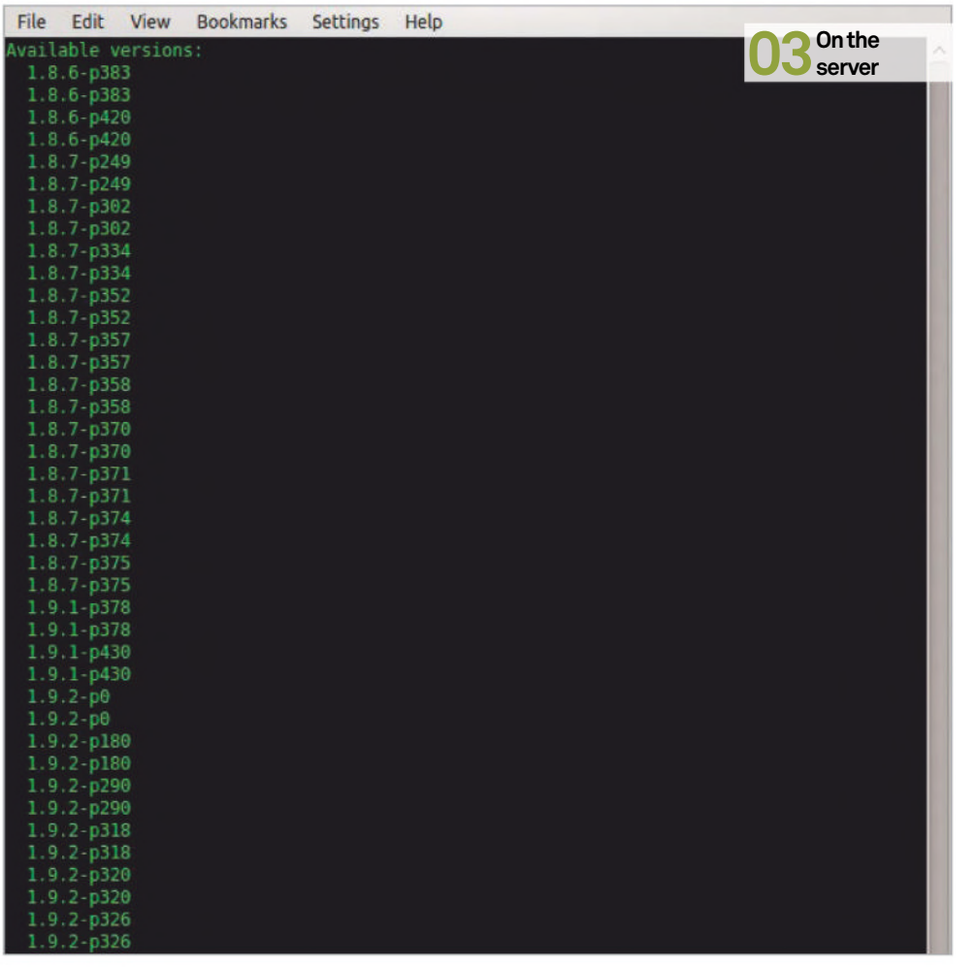
If you've been logging into your web server with a SSH password, it's time to generate keys. This is so that the login can be automated for Capistrano (or any other scripts you want to use – it's a good idea to do this on any server). You'll also need it for your GitHub account if you are going to be setting one up. Start generating the keys with:

```
ssh-keygen -t rsa
```

...if you've not got one already in ~/.ssh/ – and copy the ~/.ssh/id_rsa.pub to the server with:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub user@
remote-host
```

...which works by substituting your username, and servername or address.



03 On the server

On your servers, you'll need a similar environment of rbenv, Git and your choice of Apache or Nginx. Use rbenv to grab the latest Ruby (or an earlier one if appropriate to your app):

```
rbenv install --list
rbenv install 2.1.3
```

You won't need unnecessary packages like Ruby documentation so when you come to install bundler, specify no docs:

```
gem install bundler --no-ri --no-rdoc
```

Better yet, put:

```
gem: --no-rdoc --no-ri
```

...in ~/.gemrc. You'll also need to set up the database – or separate server - as required by your setup. A typical Rails use case is SQLite on the developer's laptop and PostgreSQL on the production servers.

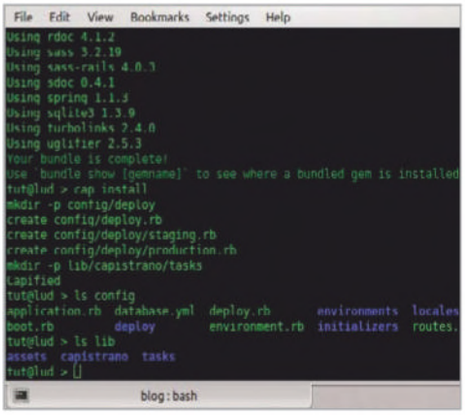
04 Bundling gems

For a Rails project, you can skip the install step and simply edit your Gemfile to include Capistrano and anything else you want. You'll want more gems. In your project's Gemfile:

```
gem 'capistrano', '~> 3.2.1'
gem 'capistrano-rails', '~> 1.1.2'
gem 'capistrano-bundler'
# if you are using RBENV
gem 'capistrano-rbenv', "~> 2.0"
# if you're using Unicorn app server
gem 'unicorn'
# Otherwise gem 'passenger' or 'puma'
```

This will also integrate bundler with Capistrano. `bundle install` downloads the gems specified in your Gemfile and installs them.

The ~> 3.2.1 means that while 3.2.2 will be installed when it's released, 3.3.0 will not to avoid major version changes breaking your code.

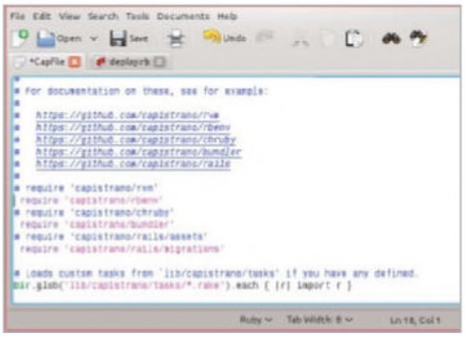


05 Config/deploy

`bundle exec cap install` creates the necessary config files and directories. It is in these that we specify the actions that Capistrano takes to deploy our app to staging or to our production servers.

Note that staging and production are created by default with files under config/deploy, and you can add a qa.rb file to that subdir if you wish to add a third stage. Alternatively, you can create all the extra stages at installation time:

```
cap install STAGES=staging,production,ci,qa
```

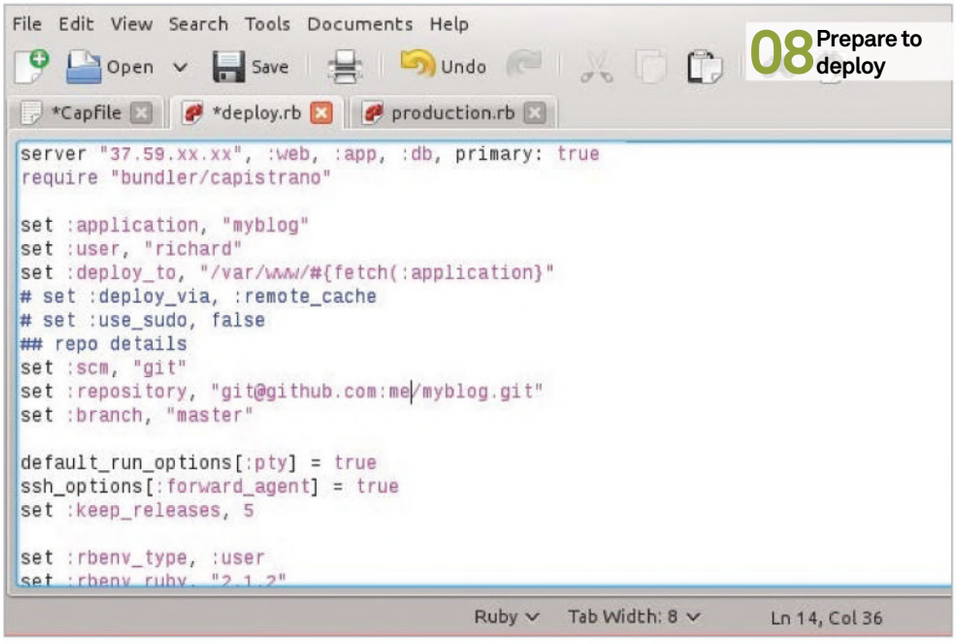


06 Config

The last step created the Capfile, tasks under lib/capistrano and deployed files under config/. Start with editing the Capfile by uncommenting the lines:

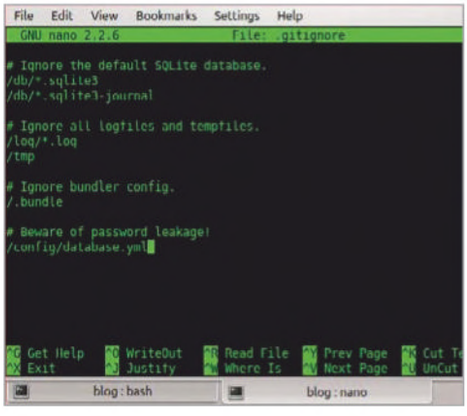
```
require 'capistrano/rbenv'
require 'capistrano/bundler'
require 'capistrano/rails/migrations'
```

...and make sure that the line starting Dir.glob at the end of the above screenshot is present and uncommented. This line ensures that all of the RB files below lib/capistrano get loaded, including any custom tasks in lib/capistrano/tasks that you'll want to later define.



08 Prepare to deploy

"Note that staging and production are created by default"



07 Secrets

Remember that everything in our app is being shared with the team and anyone else who has access to our Git repository, including exposed passwords. Edit your .gitignore file to add this line:

```
/config/database.yml
```

Now you can edit /config/database.yml for your database locally and in your staging and production environments.

Restart your Rails server and check it's working properly at http://localhost:3000

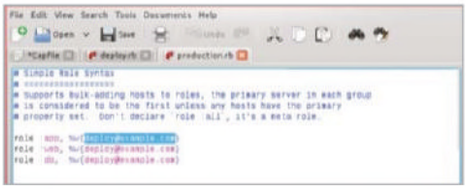
08 Prepare to deploy

In deploy.rb, ahead of `namespace :deploy`, you should add some lines relevant to your project:

```
require "bundler/capistrano"

server "37.59.xx.xx", :web, :app,
:db,primary: true

set :application, "myblog"
set :user, "richard"
set :deploy_to, "/var
www/#{fetch(:application)}"
# set :deploy_via, :remote_cache
# set :use_sudo, false
## repo details
set :scm, "git"
set :repository, "git@github.com:jsmith/
myblog.git"
set :branch, "master"

default_run_options[:pty] = true
ssh_options[:forward_agent] = true
set :keep_releases, 5
```

Note that here we're deploying to just one server. You might also want to add some specifics for your Ruby environment:

```
set :rbenv_type, :system # or :user
set :rbenv_ruby, "2.1.2"
set :rbenv_prefix, "RBENV_
ROOT=#{fetch(:rbenv_path)} RBENV_
VERSION=#{fetch(:rbenv_ruby)}
#{fetch(:rbenv_path)}/bin/rbenv exec"
set :rbenv_map_bins, %w{rake gem bundle
ruby rails}
```



09 Test and prepare

Setting tests will halt the deployment should tests fail:

```
# tests under lib/capistrano/tasks/run_
tests.cap
set :tests, []
```

Now edit the `set :server_name` and server lines in config/deploy/production.rb to your server's domain name.

We're about ready to deploy now. You could add a task deploy:setup_config to install the server software, and that would be desirable with a multiserver setup – but we're keeping things simple in this introduction, hence the earlier manual set up of the server.

10 Deploy

Now for deployment. Our config wound up the spring; to set the clockwork in motion we:

```
cap production deploy
```

For that to work first time, you'll have had to have done a bit of work on the config – beyond what we've specifically outlined in the case of many setups. Capistrano is designed to expand and stretch to cover all sorts of circumstances, and your app and server setup won't be exactly the same as anyone else's. Although, that said, if you deploy to a large cloud provider then they may have Capistrano example configs you can copy.

Capistrano's modularity and flexibility means th ere's a lot to explore to get it doing what you need. It is essentially a utility for running tasks, in parallel, across remote servers...

11 Parallel lines

With the simple on/in/do syntax of the Rake-derived DSL, flexible splitting of tasks across servers is easy. For example, to run something on every server at once:

```
on :all, in: :parallel do
  # parallel task here
end
```

Getting your servers to start a task in sequence, perhaps to avoid hitting a shared database, involves defining :sequence and then:

```
on :all, in: :sequence, wait: 15 do
  # sequential code here
end
```

For a rolling restart of a large cluster, you can group servers together to go in parallel:

```
on :all, in: :groups, limit: 3, wait: 5 do
  # Your rolling restart...
end
```

You can see the parallel execution of code for different groups of servers in the Capistrano developers' example over at **bit.ly/1FuodCh** (under 'Parallelism'), developed specifically to replace the previous version's more hacky parallel do |session|.

12 Off the Rails

There's plenty of documentation available to go beyond our simple introduction. The Capistrano website (**http://capistranorb.com**) should be your first port of call, but you'll find docs elsewhere for integrating Capistrano with everything from Jenkins to AngularJS – just make sure that you're reading something updated for Capistrano 3. Integrating Capistrano's scripts for continuous deployment into a larger environment of a continuous integration server like Jenkins is easy enough – a job in Jenkins, for example, has cap deploy as its build step (and your Git repository as its URL).

Away from Rails, gems exist to ease deployment for most web platforms, from Sinatra to Django. For Drupal (with Drush and the CapDrupal gem) you can automate taking the site offline, backing up the database, cloning your new code and pointing site root there, updating the database, and putting the site back online as well as the advantages of cap deploy:rollback.

"Away from Rails, gems exist to ease deployment"

SSHkit

The Capistrano 3 development process gave birth to the SSHkit library developed by Lee Hambley – who did much of the great work of making Capistrano 3 such an improvement over an already pretty useful piece of code.

SSHKit is a lower level toolkit that Capistrano uses for everything such as logging, formatting, connection management and pooling, parallelism, and batch execution. You too can use it for running commands in a structured way on one or more servers.

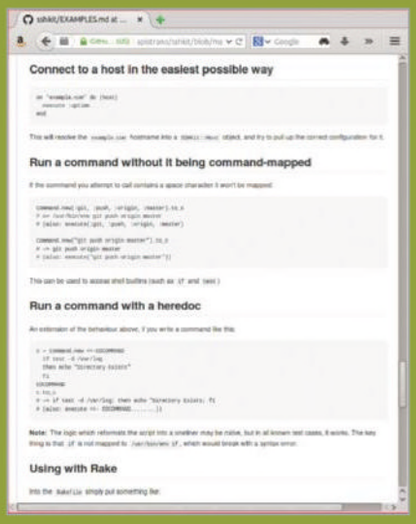
Within Capistrano, you'll find SSHkit called upon most of the time you use on(). To use it yourself, start with some of the examples on the GitHub page and adapt for your site.

Here SSHkit will build a path from the nested directories:

```
on hosts do
  within "/var" do
    puts capture(:pwd)
    within :log do
      puts capture(:pwd)
    end
  end
end
```

Behind the scenes File.join() is taking care of the slashes for you to build the paths and it should return:

```
/var/
/var/log
```



The numerous usage examples on SSHkit's GitHub page hint at its potential

About the mag

The magazine for the GNU generation

Written for you

Linux User is the only magazine dedicated to advanced users, developers & IT professionals

In-depth guides & features

Written by grass-roots developers & industry experts

Free files every issue

Four of the hottest distros feature every month - go to filesilo.co.uk to download them

subscribers to...

LinuxUser & Developer™

Try three issues for £5 in the UK*
or just $7.85 per issue in the USA**
(saving 54% off the newsstand price)

For amazing offers please visit
www.imaginesubs.co.uk/lud
Quote code ZGGZINE

Or telephone UK 0844 249 0282+ overseas +44 (0) 1795 418 661

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

YOUR FREE RESOURCES

Log in to **filesilo.co.uk/bks-835/** and download your tutorial resources **NOW!**

EVERYTHING YOU NEED TO FOLLOW THE TUTORIALS AND PROJECTS IN THIS BOOK

ubuntu®

GET THE SOFTWARE USED IN THE TUTORIALS

Python is the ultimate Raspberry Pi coding language

PACKED WITH FREE PREMIUM CONTENT

twilio™ CLOUD COMMUNICATIONS

Complete the tutorials

redhat. LINUX

Download distros

YOUR BONUS RESOURCES

ON FILESILO WE'VE PROVIDED FREE, EXCLUSIVE CONTENT FOR **LINUX & OPEN SOURCE GENIUS GUIDE VOLUME 7** READERS, INCLUDING...

- 25 distros to use in your projects including Ubuntu, BackBox 4.0, Elementary OS, OpenSUSE and many more
- 31 pieces of software to download to help you complete your Linux-based ideas including OrangeHRM and Cacti
- 6+ hours of video tutorials across 20 videos covering everything from how to write good Raspberry Pi code and an insight into Debian Linux

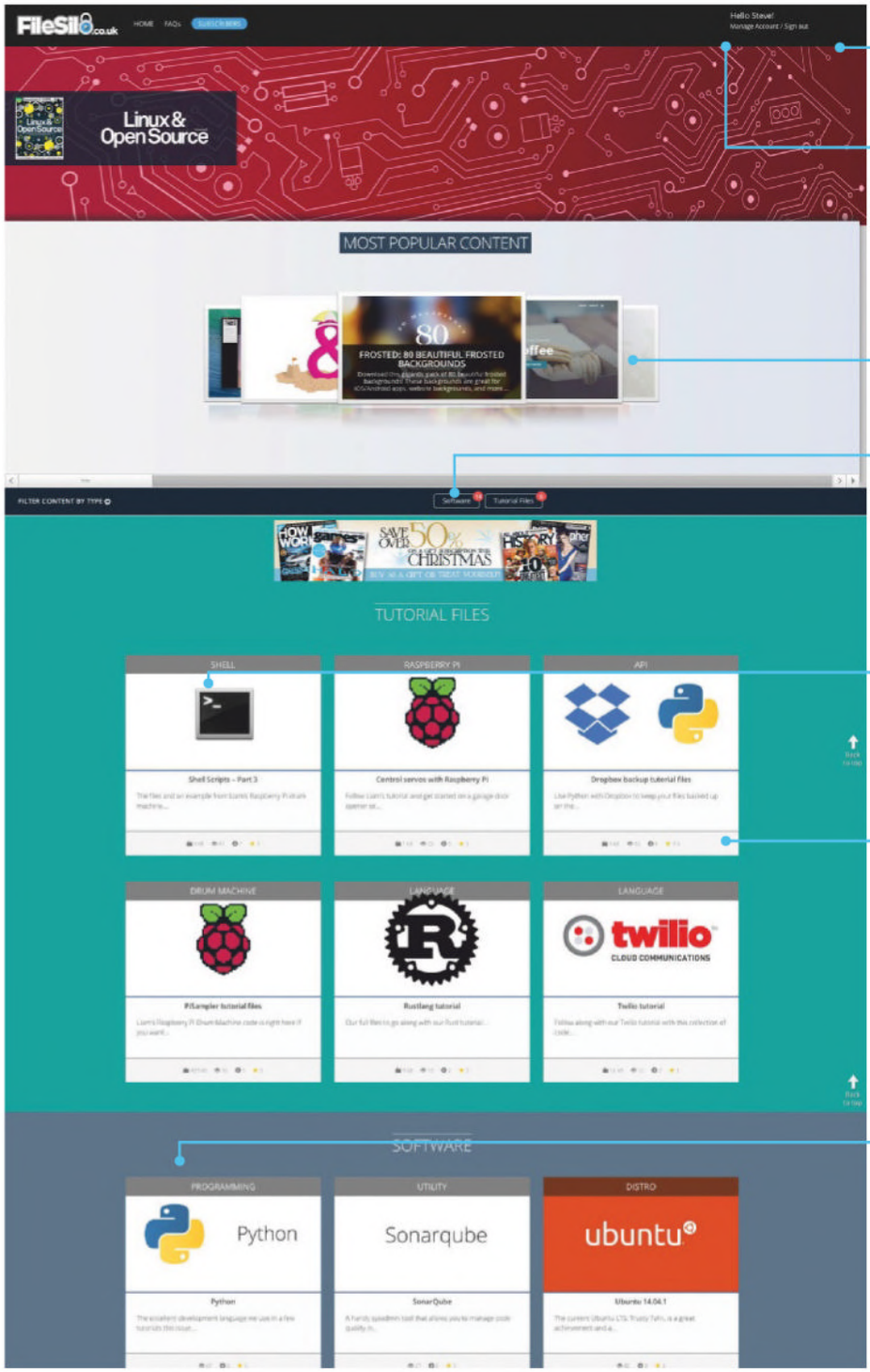
FileSilo

Go to: http://www.filesilo.co.uk/bks-835/

FILESILO – THE HOME OF PRO RESOURCES

Discover your free online assets

- A rapidly growing library
- Updated continually with cool resources
- Lets you keep your downloads organised
- Browse and access your content from anywhere
- No more torn disc pages to ruin your magazines
- No more broken discs
- Print subscribers get all the content
- Digital magazine owners get all the content too!
- Each issue's content is free with your magazine
- Secure online access to your free resources



This is the new FileSilo site that replaces your disc. You'll find it by visiting the link on these pages.

The first time you use FileSilo, you'll need to register. After that, you can use your email address and password to log in.

The most popular downloads are shown in the carousel here, so check out what your fellow readers are enjoying.

If you're looking for a particular type of content, like software or video tutorials, use the filters here to refine your search

Whether it's Python tutorials or software resources, categories make it easy to identify the content you're looking for

See key details for each resource including number of views and downloads, and the community rating

Find out more about our online stores, and useful FAQs, such as our cookie and privacy policies and contact details.

Discover our fantastic sister magazines and the wealth of content and information that they provide.

HOW TO USE FileSilo

EVERYTHING YOU NEED TO KNOW ABOUT ACCESSING YOUR NEW DIGITAL REPOSITORY

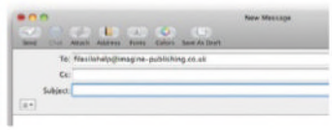
To access FileSilo, please visit http://www.filesilo.co.uk/bks-835/

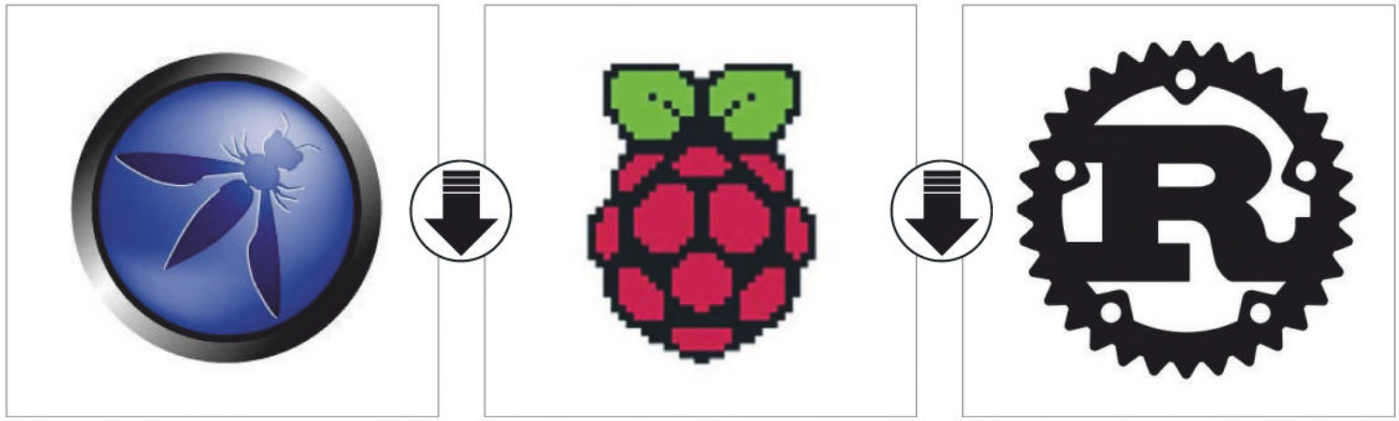
01 Follow the on-screen instructions to create an account with our secure FileSilo system, or log in and unlock the issue by answering a simple question about the edition you've just read. You can access the content for free with each edition released.

02 Once you have logged in, you are free to explore the wealth of content made available for free on FileSilo, from great video tutorials and online guides to superb downloadable resources. And the more bookazines you purchase, the more your instantly accessible collection of digital content will grow.

03 You can access FileSilo on any desktop, tablet or smartphone device using any popular browser (such as Safari, Firefox or Google Chrome). However, we recommend that you use a desktop to download content, as you may not be able to download files to your phone or tablet.

04 If you have any problems with accessing content on FileSilo, or with the registration process, take a look at the FAQs online or email filesilohelp@imagine-publishing.co.uk.





NEED HELP WITH THE TUTORIALS?

Having trouble with any of the techniques in this issue's tutorials? Don't know how to make the best use of your free resources? Want to have your work critiqued by those in the know? Then why not visit the Bookazines or Linux User & Developer Facebook page for all your questions, concerns and qualms. There is a friendly community of experts to help you out, as well as regular posts and updates from the Linux User & Developer team. Like us today and start chatting!

facebook.com/ImagineBookazines
facebook.com/LinuxUserUK

Linux & Open Source

The essential guide to mastering open source software and operating systems

The next level of Linux

Master your Linux projects with this selection of top tips and tricks. Learn how to encrypt emails, manipulate data and back up to the cloud

Encrypt and secure everything

Discover 50 ways to fix Linux, learn how to triple boot your computer, get a handle on Linux privacy, troubleshoot problems and more

Great coding projects

Utilise your Linux knowledge and create a VPN server, build a RAID array, monitor traffic and design your own games for Pebble