

**NEW**

✓Ubuntu ✓Linux Mint ✓Fedora



**FREE  
8GB** TUTORIALS,  
SOFTWARE  
& DISTROS FOR  
ALL READERS

Volume 2

# LINUX

## Tips, Tricks, Apps & Hacks

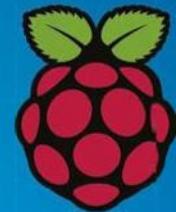
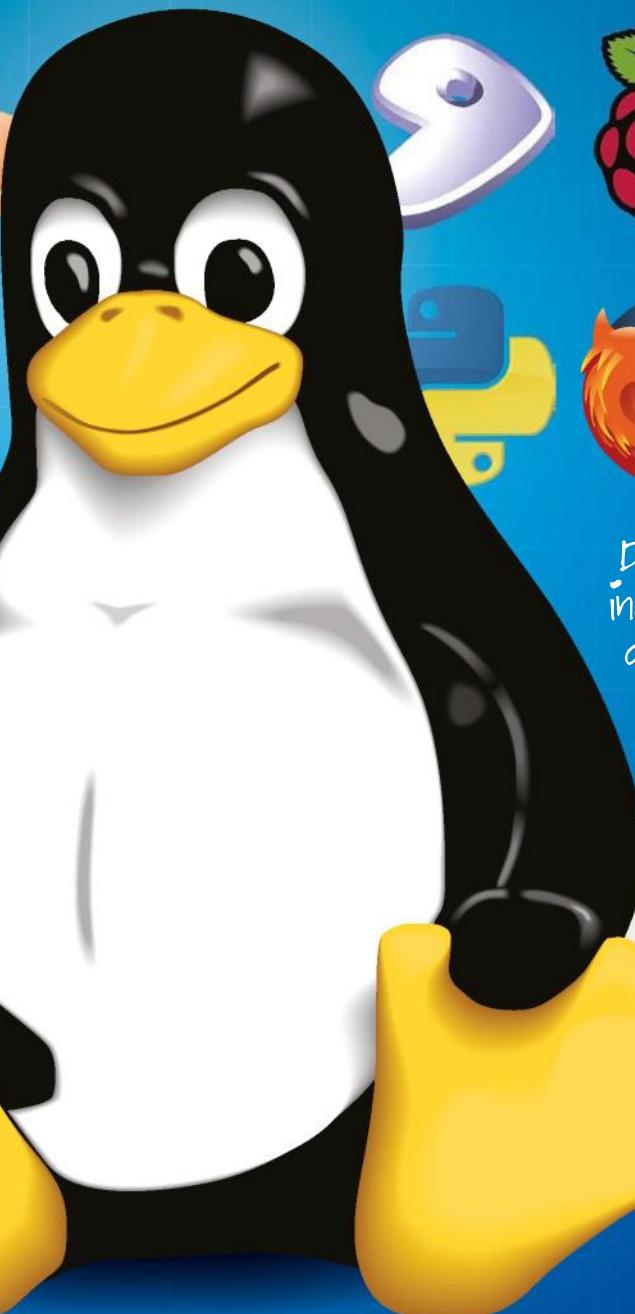
Unlock the potential of open source operating systems



Transform  
your system  
with essential  
software



Customise  
your Linux  
experience



Discover  
incredible  
distros

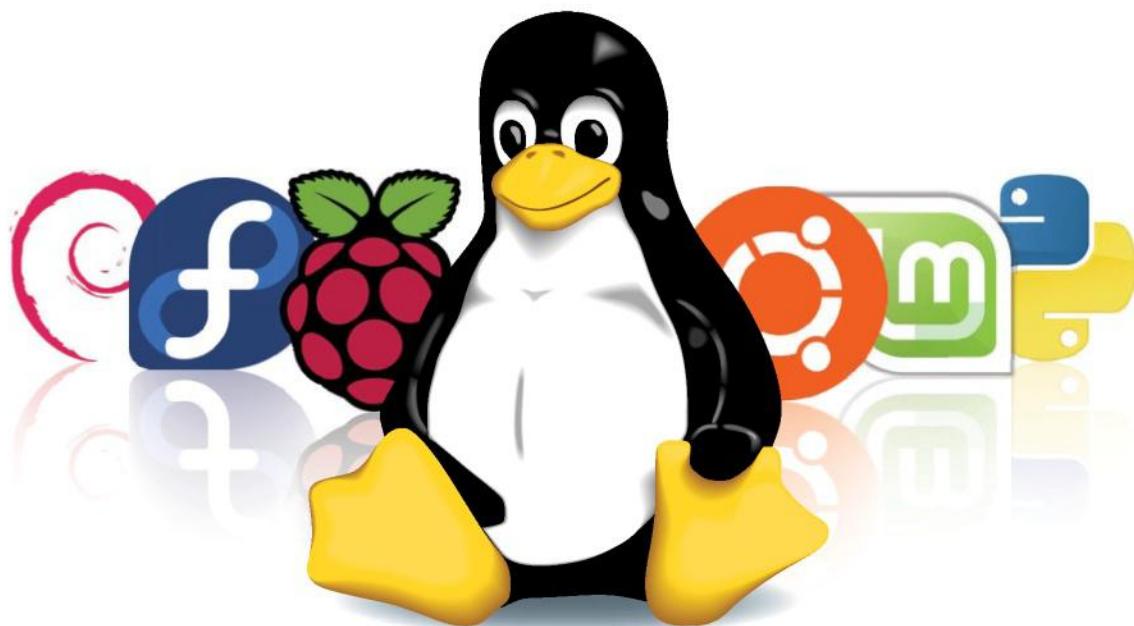


Over  
**600**  
essential  
hints and  
tips inside



# Welcome to **Linux** **Tips, Tricks, Apps & Hacks**

The revised second volume of Linux Tips, Tricks, Apps & Hacks is packed full of comprehensive features and step-by-step tutorials to help you get the most out of your Linux system. We start by looking at building your own distros, so you can have a system that works your way in no time. Whether you're using your Linux setup as a development platform, an entertainment system or even as an educational tool there's a distro to be built that's suited to your needs. The Tips section that follows includes guides to help you build, create and enhance your system – while our Tricks section features tutorials on some of the most useful free and open-source applications around and how they can improve your system. After the advanced customisation and tweaking tutorials found in the Hacks section, we review some of the best distros and apps that adhere to the FOSS philosophy.





# Linux

Tips, Tricks, Apps & Hacks

Imagine Publishing Ltd  
Richmond House  
33 Richmond Hill  
Bournemouth  
Dorset BH2 6EZ

✉ +44 (0) 1202 586200

**Website:** [www.imagine-publishing.co.uk](http://www.imagine-publishing.co.uk)

**Twitter:** @Books\_Imagine

**Facebook:** [www.facebook.com/ImagineBookazines](http://www.facebook.com/ImagineBookazines)

**Publishing Director**  
Aaron Asadi

**Head of Design**  
Ross Andrews

**Production Editor**  
Hannah Westlake

**Senior Art Editor**  
Greg Whitaker

**Designer**  
David Lewis

**Printed by**

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

**Distributed in the UK, Eire & the Rest of the World by**  
Marketforce, Blue Fin Building, 110 Southwark Street, London, SE1 0SU  
Tel 0203 148 3300 [www.marketforce.co.uk](http://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

**Disclaimer**

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

**Linux Tips, Tricks, Apps & Hacks Volume 2 Revised Edition © 2015 Imagine Publishing Ltd**

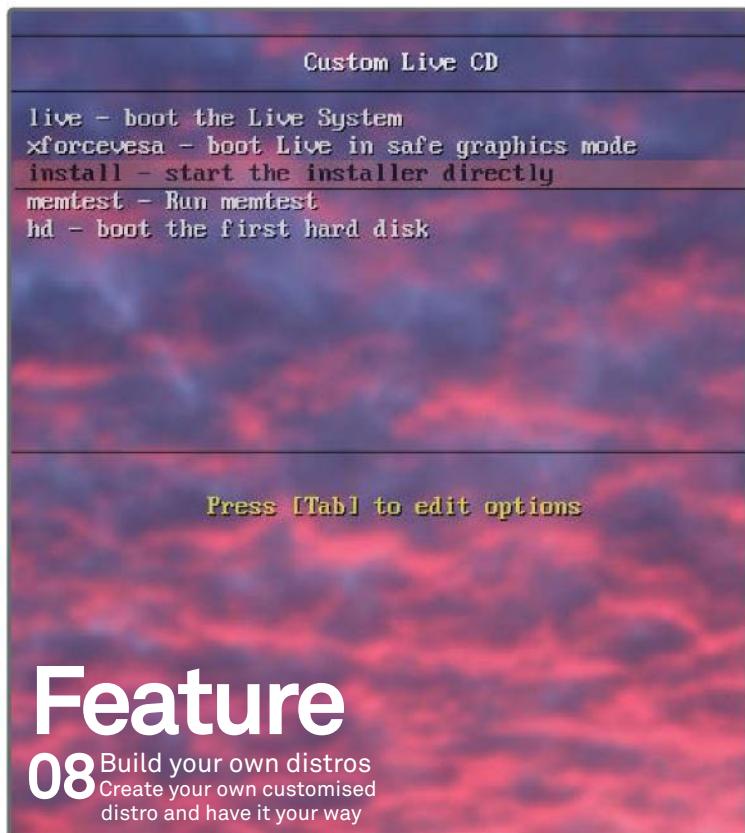
ISBN 978-1910 439 791

Part of the

**LinuxUser**  
& Developer  
bookazine series



# Contents



# Tips

- |    |  |    |                                       |
|----|--|----|---------------------------------------|
| 18 | Get started with system administration     | 40 | Host your own webmail server          |
| 24 | Test your network's security               | 44 | Deploy Fedora over a network          |
| 28 | Protect your network                       | 48 | Make your own DEB and RPM packages    |
| 32 | Configure a secure virtual private network | 52 | Dual-boot from an external hard drive |
| 36 | Build your own pro-grade firewall          | 54 | Run Linux on an Android device        |

# Tricks

- 

**62** Network wirelessly with wicd

**66** Manage your system with Webmin

**70** Synchronise your files with Unison

**74** Make a small business database with LibreOffice

**78** Create and save data with a MongoDB database

**82** Maintain and manage all of your machines with Puppet

**86** Visualise directory structures with Graphviz

**90** Edit videos in Kdenlive

**92** Build your own private cloud with ownCloud

**96** Design exciting presentations with Hovercraft

**100** Set up a wireless access point with a Raspberry Pi

**“Find the best distros for your needs”**

**“Find the best distros for your needs”**





## Hacks

- 104** Turbocharge your cloud
- 110** Speed up Linux with Openbox
- 114** Bypass restrictive firewalls using SSH tunnelling
- 118** Create a custom build of Gentoo
- 122** Create a custom Linux kernel to optimise performance
- 126** Resize your disks on the fly with LVM
- 130** Scrape Wikipedia with Beautiful Soup

## Apps

<b>136</b> openSUSE 13.1 RC 1	<b>146</b> Ubuntu 14.04 LTS	<b>153</b> SpiderOak
<b>138</b> Linux Mint 16 RC	<b>148</b> Geany	<b>154</b> Openshot
<b>140</b> Tails 1.2	<b>149</b> Eclipse	<b>155</b> Kdenlive
<b>142</b> Fedora 19 Schrödinger's Cat	<b>150</b> wattOS R8	<b>156</b> Clementine
<b>144</b> LXLE 14.04	<b>152</b> Dropbox	<b>157</b> Banshee

# BUILD YOUR OWN DISTRO

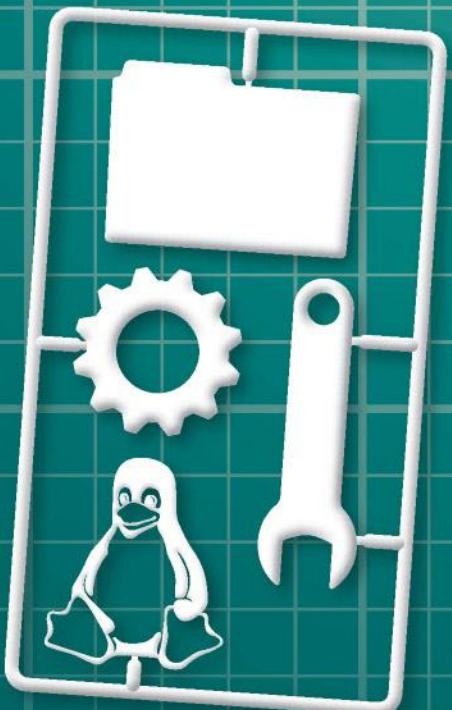
Discover the different methods available for creating your own customised distro and have everything working your way in no time



**There are a few reasons why you might want to build your own distribution.** You might want to build a custom install CD to match the policy of your organisation. For example, a GNOME desktop with Chrome as the web browser might be the standard desktop where you work. That touches on another motivation for wanting to create a customised installer: sometimes the creator of the distribution makes a decision that you simply don't like. Canonical's decision to switch to its own UI, Unity, ranks amongst its most controversial decisions. However, by using some of the methods that we explore here, you could create a distribution that is standard Ubuntu, but with a traditional desktop that you are more comfortable with.

There are other, niche reasons for wanting to build your own distribution. You might need to put something small and lightweight together for an older computer. You might need to build a live media ISO that you are able to carry around with you and to bring your favourite set of tools to bear when you need them.

The methods of creating a custom distribution are varied, but they can be divided into two main categories: you can modify a running distribution and then distribute it, or you can modify the installation ISO (called 'remixing') so that it installs your modified distribution in the way you have specified. We're going to take a look at four approaches.



# Remastersys

Remastersys is a tool that extracts the configuration from a running Ubuntu or Debian installation and then turns this into an installable ISO image. This means that you carry out the customisation using the standard tools that you normally use, such as the package management system and GUI configuration tools. When you have everything set up the way you want it, you can clone the system and deploy it. Additionally, you can use Remastersys to make a clone of a working system.

## Fetch Remastersys

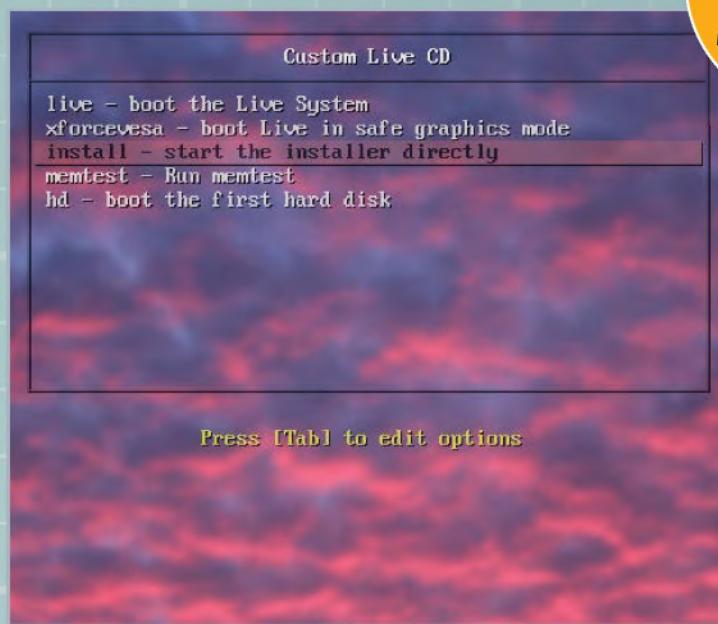
The development status of Remastersys is currently in transition. At time of writing, the best policy is to visit the Remastersys website and to cut and paste the repository details from there. For example, if you are using Ubuntu 13.10, download the GPG key and add it from the command line with:

```
command line with  
wget -O - http://www.remastersys.com/  
ubuntu/remastersys.gpg.key  
sudo apt-key add remastersys.gpg.key  
then add...  
deb http://www.remastersys.com/ubuntu  
precise main  
...to the end of /etc/apt/sources.lst by invoking a  
text editor as root
```

Following that, type `sudo apt-get install remastersys remastersys-gui` in order to install Remastersys and its GUI.

## Using Remastersys

When you have the installation set up the way you want it, launch Remastersys by typing `sudo remastersys-gui`. The first option we need to visit is the customisation page which is reached by clicking on the Customise button. From here, you can change branding options such as the various splash images. From within this page, click on Copy Settings. This takes you to a further page on which you can select the user whose

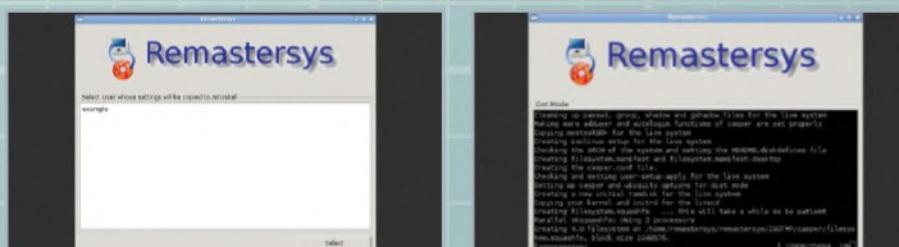


## ■ Booting from the installation ISO

**“ The future of the GUI portion of the project seems less certain ”**

settings will be copied to `/etc/skel/`. In other words, these are the settings that will become the defaults for all new users on the new system. If you skip this stage, new users will simply have the default settings for the distribution.

Finally, build the installation ISO simply by clicking on the Distribution button on the main menu page. The ISO is deposited into the /home/remastersys/ folder. Use networking to transfer the ISO file to the outside of the VM. We usually install Filezilla and transfer to a local FTP server. You can now boot the ISO on the target machine and carry out a regular Ubuntu installation.



## Copying the skeleton information for new users Building the ISO

# The future of Remastersys

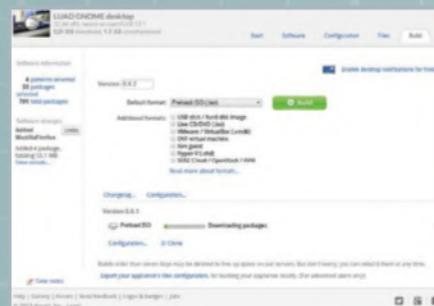
The long-time developer of Remastersys recently decided to give up development. Fortunately, he has chosen to release the source code so that other developers can take up the mantle. The future of the GUI portion of the project seems less certain, but Remastersys is also fully functional from the command line. For the moment, the binaries of Remastersys are still freely available from the developer's web site ([www.remastersys.com](http://www.remastersys.com)).

The situation is constantly in flux, so search around for the latest forks. The System Imager project, which uses Remastersys, is a good source of up-to-date information, and can be found here: <http://system-imaging.blogspot.co.uk>.

# SUSE Studio

**SUSE Studio** allows you to build a customised SUSE Linux installation using a web interface. Although it's easy to use, that doesn't mean it has compromised on options.

Initially, you choose a base template such as KDE Desktop or Server. From this point, you begin the configuration properly. The first tab is labelled Software, which is where you choose software packages with an interface that is categorised and searchable.



## Example deployment: Business desktop

Here we're going to put together an example appliance. In this case, the appliance will be a business desktop that based around GNOME. We'll add a few customisations as we go along, and we want to finish up with an installable ISO that we can use for deployment.

Begin by setting up an account on the SUSE Studio website (<http://susestudio.com>). You do this by following the 'Sign In Or Create An Account' link on the front page, and it is possible to use one of your existing social networking accounts such as Facebook or OpenID if you prefer.

Once you have an account, click on 'Create New Appliance...'. On the next screen, choose the GNOME Desktop base template, making sure that you are selecting from the templates that

**“You are able to add extra repositories and even custom RPM packages”**

- Pros**
  - Could hardly be easier to use, sharing of appliances is built into the site
- Cons**
  - Build speed varies, you might hit a wall with really complex customisations

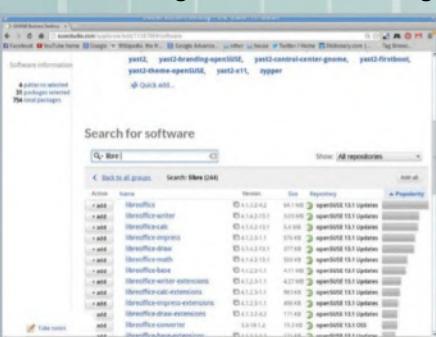
Waiting for the ISO to build



## Tip

You can upload RPMs that aren't in the standard repos using the software page

Adding in some custom branding



Selecting software packages to begin customising the desktop



Booting from the installation ISO

The screenshot shows the SUSE Studio configuration interface. On the left, there's a sidebar with 'Software information' (4 patterns selected, 32 packages selected, 787 total packages), 'Software changes' (Added libreoffice, Undo), and a note about adding 33 packages totaling 235.4 MB. The main area has tabs: General (selected), Personalize, Startup, Server, Desktop, Appliance, and Scripts. The General tab contains sections for 'Default locale' (Language: English (UK), Keyboard Layout: English (UK)), 'Default time zone' (Region: Europe, Time Zone: United Kingdom), 'Network' (options: Do not configure network, Configure network during first boot, Use NetworkManager to configure the network at run-time, Discover network settings automatically (DHCP) (selected), Manually configure network), and 'Firewall' (checkboxes: Enable firewall, Open SSH port (22), Open HTTP ports (80, 443)). There's also a 'Users and groups' table:

Login	UID (optional)	Password	Group	Home directory	Shell
root	0	password	root	/root	/bin/bash
killermike		password2	users	/home/killermike	/bin/bash

A yellow tip bubble says: 'Tip: The build process and download speed vary according to server load so get things ready before needed'.

We'll leave the network options as they are, but this is where you would disable DHCP and specify a static IP address for the workstation, or disable the firewall if you needed to. At the bottom of the page, we can see a list of users and groups. It's a good idea to change the root password from the default. Now click on 'Add new user...' and create a standard user who is a member of the Users group.

Moving to the Personalize sub-tab of the Configuration page, we can now add some custom branding. This might fit in well with the policies of your organisation, and it is also extremely handy for at-a-glance identification of a desktop within a busy IT environment.

The Files tab is worth a visit if you need to add custom files to the distribution. You can add single files or archives. For example, if you wanted to add a file to the desktop of every new user, you should upload it and specify that it should be placed in '/etc/skeleton/Desktop'. If you wanted to place a file within the home directory of the user that you have created called John, add it to '/home/john/'.

As a finishing touch, pop into the Configuration>Desktop page. Tick the box to

automatically log the user in. Add the command firefox to the Autostart desktop user log-in section to automatically start Firefox. Opinions vary, but these options allow the user to get straight to work.

## Build the ISO

The options within the Build tab are particularly interesting because they allow you to specify the output format of your custom build. This means that you don't necessarily have to carry out a full installation in order to use your custom build.

For example, you can create a virtual machine that will directly boot within a visualiser. If you want to work like this, you will probably need to skip back to the Appliance sub-tab within the Configuration tab to define the parameters of the VM. Here, you can choose options such as allocated memory and set up the LVM partition arrangement. Apart from the various VM environments you can directly create, you can also create a traditional ISO installer, a hard disk image or a live CD/USB image.

In order to create a traditional installation ISO, select 'Preload ISO (.iso)' in the Default format and click on the 'Build' icon. This can take a few

This screen shows us configuring details such as the users and network settings in the Configuration>General page

The screenshot shows the SUSE Studio login page. It has tabs for Software information, General, Personalize, Startup, Server, Desktop, Appliance, and Scripts. The General tab is selected. It contains sections for 'Default locale' (Language: English (UK), Keyboard Layout: English (UK)), 'Default time zone' (Region: Europe, Time Zone: United Kingdom), 'Network' (options: Do not configure network, Configure network during first boot, Use NetworkManager to configure the network at run-time, Discover network settings automatically (DHCP) (selected), Manually configure network), and 'Firewall' (checkboxes: Enable firewall, Open SSH port (22), Open HTTP ports (80, 443)). There's also a 'Users and groups' table:

Login	UID (optional)	Password	Group	Home directory	Shell
root	0	password	root	/root	/bin/bash
killermike		password2	users	/home/killermike	/bin/bash

The SUSE Studio login page

minutes to complete, depending on how large and complicated your custom image is. Although it may take several minutes for your image to build, once built, your appliances remain on the site and can be downloaded without delay. The final tab, Share, is an intriguing function that allows you share your finished appliances with other people.

Boot the finished ISO as you would any other installation ISO. Confirm that you wish to erase all data on the hard disk when prompted.

# Ubuntu minimal installation



## Hard drive

If you wish you can add a virtual hard drive to the new machine. You can either create a new hard drive file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard drive is **8.00 GB**.

- Do not add a virtual hard drive
- Create a virtual hard drive now
- Use an existing virtual hard drive file

android.asus.vdi (Normal, 8.00 GB)

< Back      Create      Cancel

Creating a blank hard disk image in VirtualBox

**Canonical provides a minimal Ubuntu install CD.** It's smaller than the regular installation ISO and it installs a minimal version of the distribution. At its most basic, it gives the user a command line, network connectivity and not much else. From this bare-bones beginning, it's possible to selectively add components while leaving out most of the cruft that tends to come with a standard distribution.

We're going to work from within a virtual machine for safety and convenience. In our case, we're going to use Oracle VirtualBox but any of the major virtualisers will work. Once we have it set up the way we want it, we can use Remastersys to turn it into an ISO that can be distributed. We can then transfer this ISO from within the VM to an FTP server.

## Example deployment: Minimal Openbox Desktop

Fetch the installation media from <http://tinyurl.com/ygawub> and create a new virtual machine. 512MB is a sensible minimum when allocating memory, but more memory can also help greatly with speeding things up. An 8GB hard disk file should be adequate for most people's requirements. It's usually worth allocating as many CPU cores to the VM as you can.

**Pros**  
Excellent way of keeping the distro standard yet minimal too

**Cons**  
Time consuming to carry out from start to finish

**You'll usually find it's worth allocating as many CPU cores to the VM as you can**

## Begin the customisation

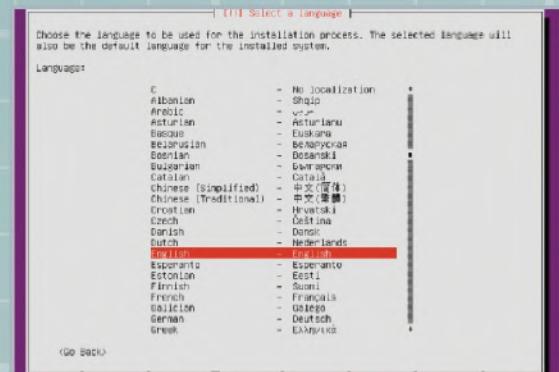
Once you have booted the ISO from within the VM, begin by filling in the localisation details using the text mode interface. Next, the installer will attempt to find your network using DHCP. Following the network detection phase, fill in a hostname that will be used to identify this computer on the network. Once you've done this, select a mirror that is geographically close to your location.

The installer should now begin to download packages. Once the packages have come through the network, set up the username for the standard user. You should be able to use common sense to ask the question that comes next, regarding your time zone and default user and password.

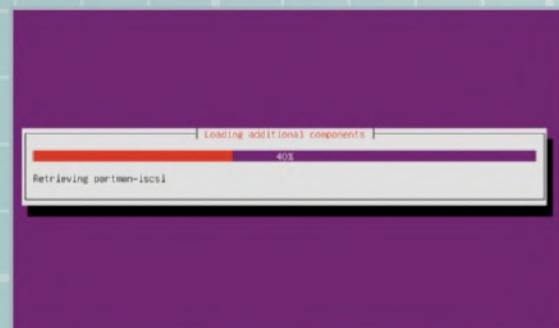
**Tip**  
Add the kernel extensions for your virtualiser to enable things like cut and paste between host and guest

## Clonezilla

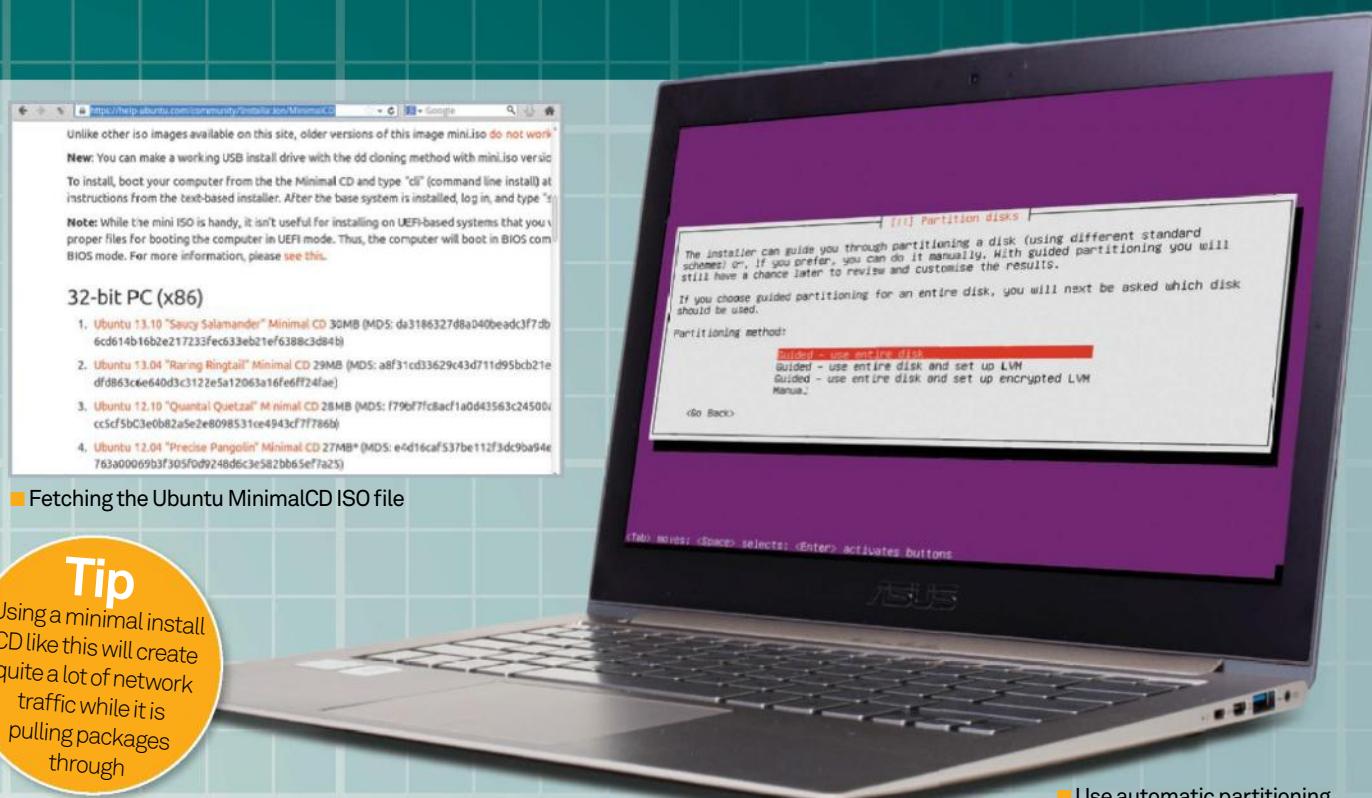
Clonezilla (<http://clonezilla.org>) is a live CD that can be used to make complete system backups. It uses an algorithm that avoids copying the empty space on a hard disk and produces files that are as small as possible. This could be used as an alternative method to distribute a customised distribution as a hard disk image.



Starting a minimal installation



Fetching the initial set of base packages



#### ■ Fetching the Ubuntu MinimalCD ISO file

### Tip

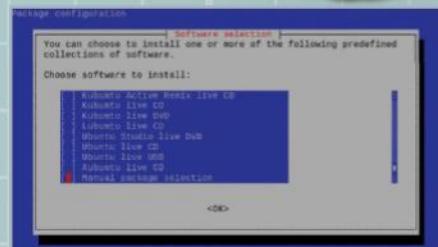
Using a minimal install CD like this will create quite a lot of network traffic while it is pulling packages through

When prompted, allow the installer to allocate the disk partitioning by selecting the 'Guided - use entire disk' option. Confirm that you want to write to the disk when prompted. The actual layout that you use now isn't important as we will be producing an ISO that will carry out the installation of our custom distribution from scratch. Once the partitioning has completed, the installer will fetch the packages need for the base installation and begin installing them.

### Customisation decisions

When the base installation is complete, you will be presented with the Software selection menu. At this point you have to make a decision. If you want to, you can select one or more of the provided templates. For example, you could select Kubuntu desktop option and have a fairly complete desktop system from the beginning. There are other options to establish a LAMP web server or a Mythbuntu media system, and many others. Most of the rest of this tutorial assumes that you don't select any of these options so that we can customise completely from scratch.

More downloading and installation follows. Confirm that you want GRUB installed to the MBR when asked. This brings us to the end of the initial installation phase. Eject the ISO and reboot the VM when prompted.



#### ■ Tasksel helps assign a specific role to your distro

### First reboot

Upon booting the minimal installation for the first time, you should be prompted for your username and password. We can now start to customise the system. Install X.org server and Openbox window manager (feel free to substitute another WM/DE if you prefer) by typing `sudo apt-get install xorg openbox`. When this has completed, type `startx` to test the GUI. Click on the backdrop to bring up a menu that will allow you to launch a terminal window. Now you can begin customising the system. Make things as comfortable as you like, but remember that anything that is installed on this system will end up on the target system. `sudo apt-get install firefox synaptic lxterminal mousepad lxdm` will install and set up the Firefox web browser, Synaptic (GUI package manager), LXTerminal (more

#### ■ Use automatic partitioning

### Tasksel

Tasksel is a system that installs and configures a series of packages related to a particular 'role' such as that of a web server, a full KDE desktop or a media workstation. You might be prompted for task selection during an install, but you can access this feature on a running installation at any time.

Add the tasksel package with `sudo apt-get install tasksel`, and then type `sudo tasksel` to run it. You'll be presented with the familiar text-mode interface, space to select an option and tab to switch fields. Naturally, you can add multiple tasks.

comprehensive terminal application), Mousepad (a GUI text editor) and LXDM (graphical login manager). That little lot will add about 30MB to the installation ISO that you will create, and about 100MB on the hard disk.

What you actually add is up to you. Apart from adding packages, you can add desktop customisations such as changing the backdrop. When you've got things just how you want them, create a distribution medium using Remastersys or a disk cloning tool such as Clonezilla.

# Ubuntu Builder

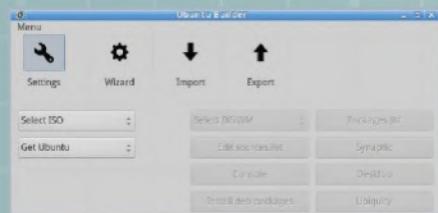
**Ubuntu Builder** is a GUI application that allows you to take the contents of a standard Ubuntu installation ISO and modify it to create a new, customised ISO for redistribution. It's a fairly simple application, however, and not designed for deep modifications of the type that some of the other methods allow.

Ubuntu Builder is a standalone application that runs on your desktop, and it even runs on distributions other than Ubuntu. It works by modifying a standard Ubuntu installation ISO, downloading and inserting or removing packages for you.



## 01 The installation

Start by adding the Ubuntu Desktop PPA by typing `sudo add-apt-repository ppa:f-muriana/ubuntu-builder` into a terminal. Now run `sudo apt-get update` followed by `sudo apt-get upgrade` to update the package lists on the system. Use `sudo apt-get install ubuntu-builder` to carry out the installation.



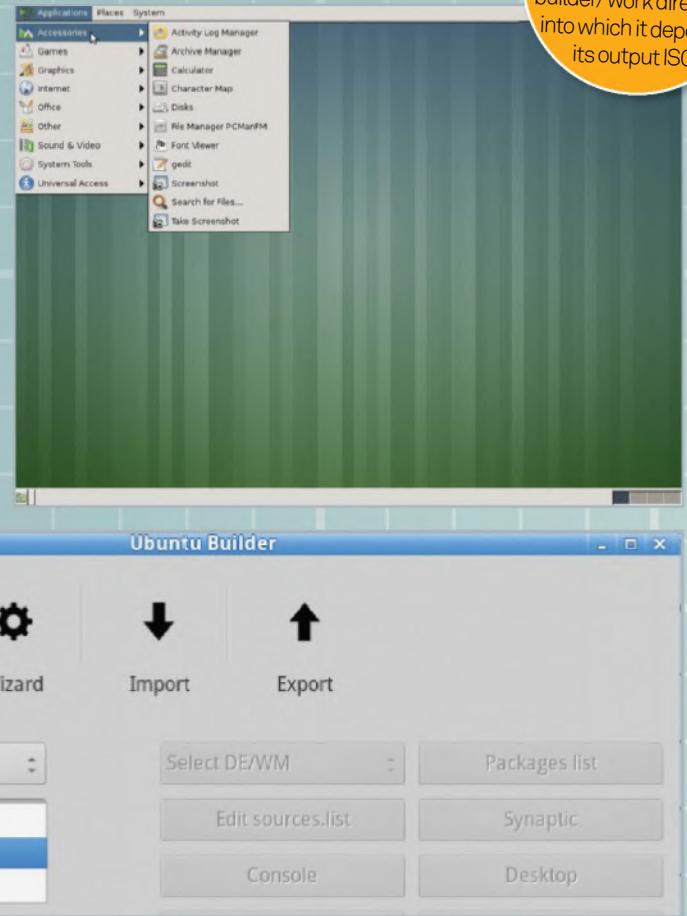
## 02 Launch Ubuntu Builder

You can now launch Ubuntu Builder by typing `sudo ubuntu-builder` in the Terminal or by clicking on its launcher icon in the launcher menu. At this point you should be able to see the basic root interface.

**Pros**  
This is a nice, simple tool - there's little you can do to make this all go wrong

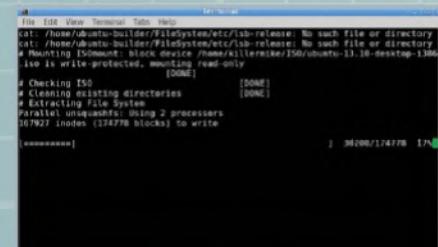
**Cons**  
Not a huge amount of customisation depth. Lacked polish and felt a bit buggy in use. ISO build process is also a bit slow

The end result:  
Ubuntu 13.10 with MATE, a more traditional desktop



## 03 Fetch ISO

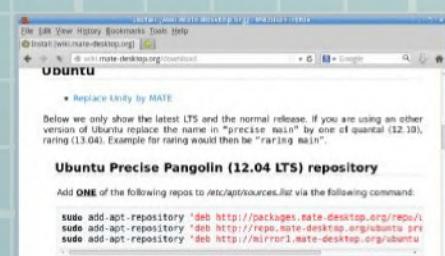
You can fetch the current ISO by clicking on the 'Get Ubuntu' button in the main interface. However, it's worth mentioning that we actually found manually fetching the latest standard install ISO from the Ubuntu website to be more reliable.



## 04 Select and unpack the ISO

Point Ubuntu Builder to the standard installation ISO by clicking on the 'Select ISO' button. This should invoke the unpack procedure in a Terminal window, enabling us to modify the contents of the ISO. Wait for this process to finish.

**Tip**  
Ubuntu builder creates a /home/ubuntu-builder/work directory into which it deposits its output ISO



## 05 Add the MATE repository

Click on the 'Edit sources.list' button. This opens a text editor. Cut and paste the appropriate repository line (beginning with deb) from the MATE installation guide ([wiki.mate-desktop.org/download](http://wiki.mate-desktop.org/download)).

```
File Edit View Terminal Tabs Help
Reading state information...
0 upgraded, 0 newly installed, 0 to remove and 268 not upg
Reading package lists...
Building dependency tree...
Reading state information...
The following packages will be REMOVED:
  lightdm lightdm-remote-session-freerdp*
  lightdm-remote-session-uccsconfigure* ubuntu-desktop*
0 upgraded, 0 newly installed, 4 to remove and 267 not upg
After this operation, 734 KB disk space will be freed.
(Reading database ... 173432 files and directories currentl
Removing ubuntu-desktop ...
Removing lightdm-remote-session-uccsconfigure ...
Purging configuration files for lightdm-remote-session-ucc
Removing lightdm-remote-session-freerdp ...
Purging configuration files for lightdm-remote-session-fre
Removing lightdm ...
Purging configuration files for lightdm ...
Removing user lightdm ...
Warning: group 'lightdm' has no more members.
Done.
Processing triggers for man-db ...
Processing triggers for ureadahead ...

```

## 06 Select MATE as the desktop environment

Click on the Select DE/WM button. In the menu, select MATE as the desktop. This should invoke a Terminal screen while the packages are replaced. Allow this process to finish.

```
File Edit View Terminal Tabs Help
STARTING BUILDING PROCESS
# Checking configs [DONE]
# Checking folders [DONE]
# Loading configs [DONE]
# Deleting obsolete files [DONE]
# Setting up distribution informations [DONE]
# Cleaning up the build environment [DONE]
# Copying boot files [DONE]
# Creating File System
Parallel mksquashfs: Using 2 processors
Creating 4.0 filesystem on /home/ubuntu-builder/ISO/casper/
block size 131072.
!
```

## 07 Create the remixed ISO

Click on the 'Build' button at the top of the main window. This will open a Terminal window that displays the progress of the ISO build process. This might take a long time (an hour or more) depending on the speed of your machine.

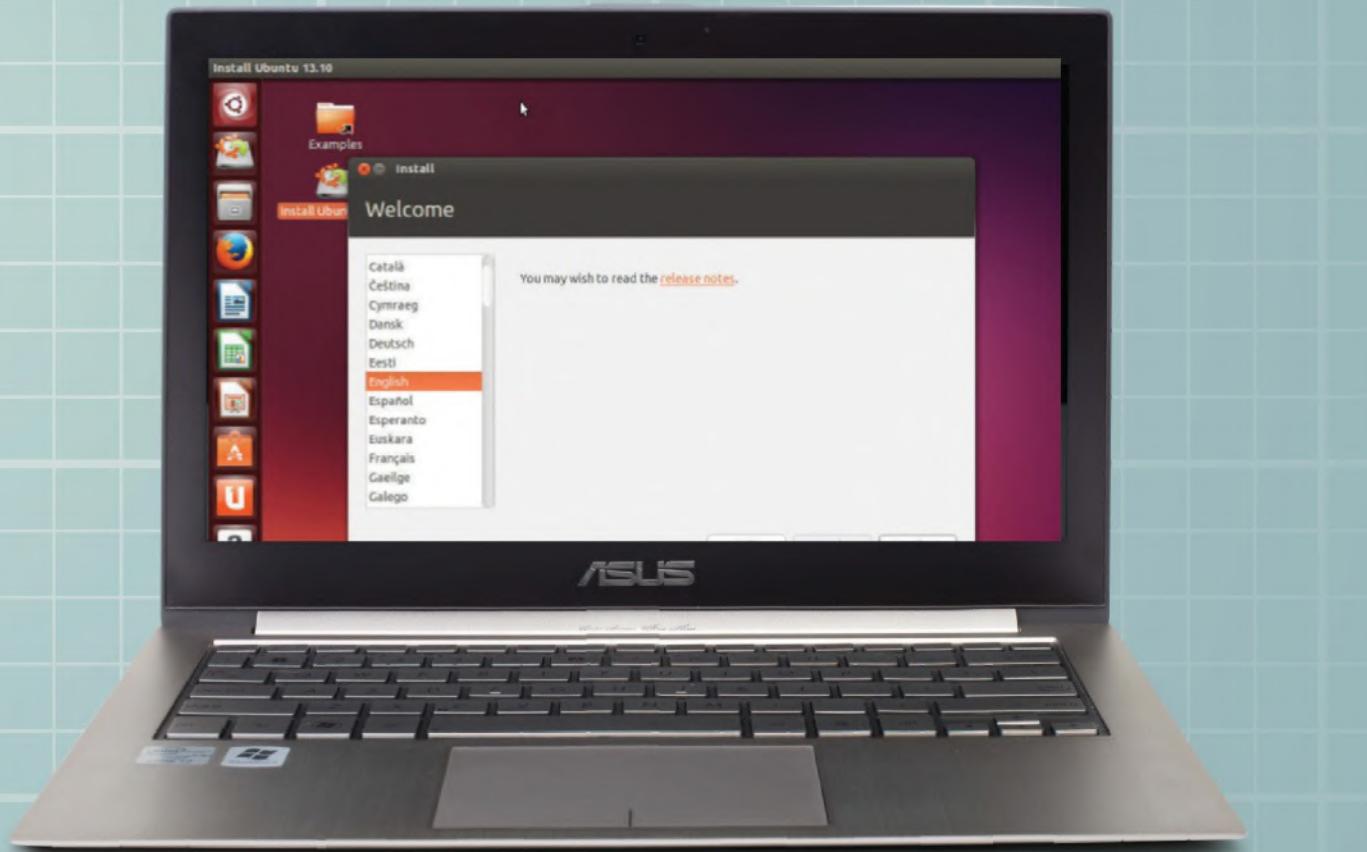


### Tip

Use the Synaptic button to reinstall LightDM when changing desktop environment

## 08 Install the ISO

Use the installation disk in the same way that you would normally install Ubuntu... and that's all it takes! You're now ready to start using your new, customised distribution. Enjoy!



**Ubuntu Builder is not designed for deep modifications that some other methods allow**

# TIPS

## Build, create and enhance your system

### 18 Get started with system administration

Unlock the full potential of Linux while learning how to manage it

### 24 Test your network's security

One of the best ways to test your security is to try to tear it apart...

### 28 Protect your network

Build a gateway server that can intelligently filter content

### 32 Configure a secure virtual private network

Stop worrying about SSH vulnerabilities and careless users

### 36 Build your own pro-grade firewall

Create a multi-network firewall with a redundant computer

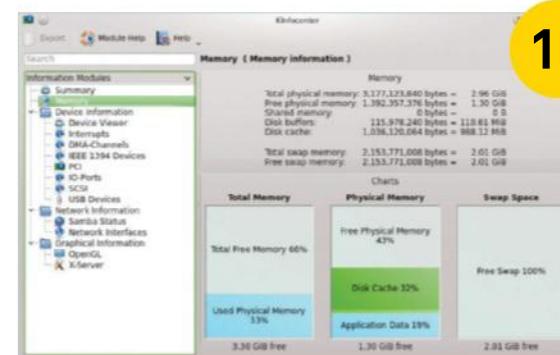
### 40 Host a webmail server

Manage your own webmail server for personal accounts

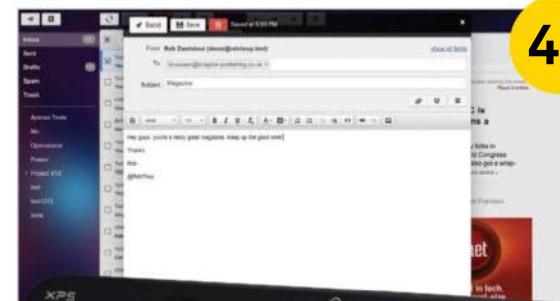
### 44 Deploy Fedora over a network

Learn how to install Fedora to an entire LAN

“The advantages of installing and running Linux on an Android device are manifold”



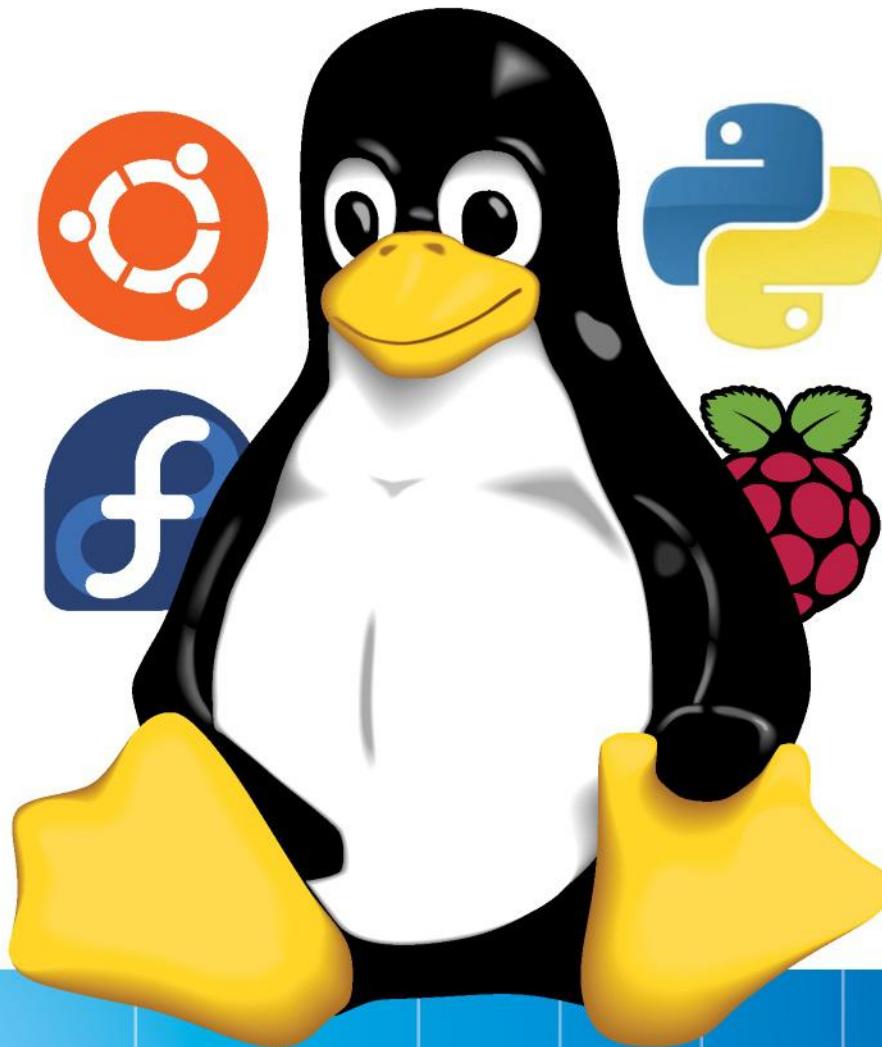
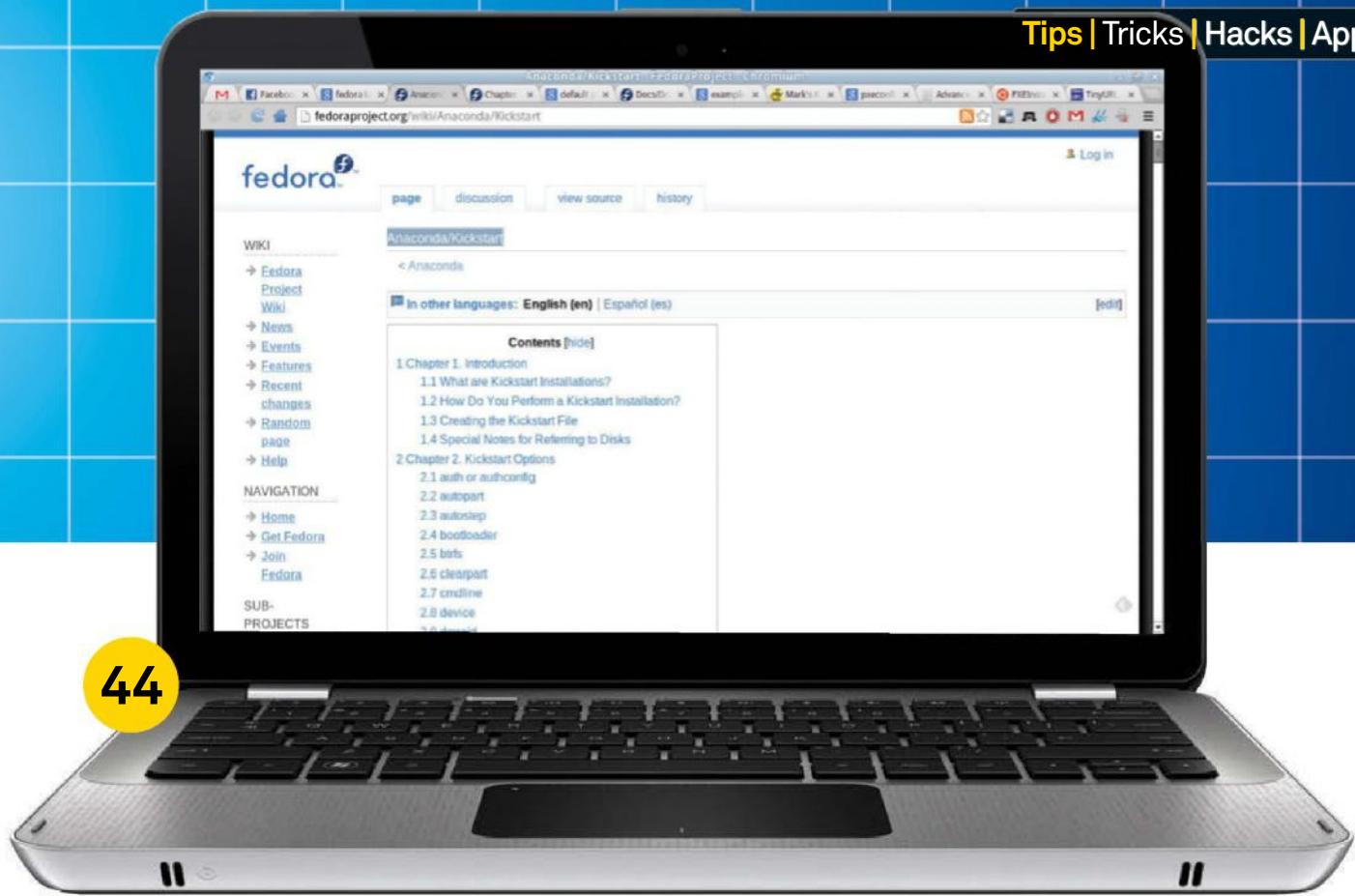
18



40



36



**48 Make your own DEB and RPM packages**  
Manufacture the two most common types of Linux package

**52 Dual-boot from an external hard drive**  
Get a multi-booting hard drive you can connect to any computer

**54 Run Linux on an Android phone or tablet**  
Get an ultra-portable version of Linux on your phone

# Get started with system administration

Unlock the full potential of Linux while learning how to manage it effectively...

**Linux is the operating system that has more network card drivers than video card drivers, if you catch our drift.** Linux was made for network. Granted, it's not too shabby in other areas, but it really excels in the networked environment. Today Linux powers most of the world's servers, whether on the internet or an intranet. One of the core competencies of Linux, which has made it perfect for running servers and services, is its

system administration features. These aren't just useful for servers in multimillion-pound companies, but even if you're using Linux at home. They give you a very smart and efficient way to control and optimise your system to your exact requirements. This article is designed to teach you about Linux system administration from a beginner's point of view. Most of the tasks we will cover can be carried out by readers who

are relatively new to 'getting their hands dirty', but we'll also cover a good few advanced tips for those who want to delve a little bit deeper.

**"Linux powers most of the world's servers"**

## KEY

**\$** = regular user commands

**#** = root user commands

The latter must be used as root or by using the sudo command.

## Advanced Tip:

If you are looking for single sign-on for the applications and services, you should look into Linux pluggable authentication modules (PAM). PAM provides a plug-in like architecture to develop authentication back-ends. There are many PAM modules in existence, such as FTP, OpenPGP smartcards etc. You can see the complete list of available modules at [www.linux-pam.org/modules.html](http://www.linux-pam.org/modules.html). This will save you lots of time creating individual users and your users will enjoy the freedom of using their existing credentials instead of remembering new ones.

## The tools

In this section we'll look at doing things using some of the tools designed to help us in system administration tasks.

### Managing users

While installing Linux you are asked to create at least two users for the system. One is root, which has the ultimate power over the system, and the other one is the regular user – restricted to performing day-to-day tasks. Let's see what else is possible with regards to users.

#### To add a user:

**# adduser <username>**

On some systems (such as Ubuntu) you will also be asked enter the password for the new user. On other systems you will need to create passwords separately:

**# passwd <username>**

The passwd command can also be used to change other users' passwords. When not used with a username, it offers to change the password for the user issuing the command.

### Installing packages

Most Linux distributions use either the Debian package format (DEB) or Red Hat Package Manager (RPM). As already evident by the package format name, DEB is used on Debian-based distributions such as Ubuntu and Knoppix, while RPM is used on Red Hat

Linux-based distributions such as Fedora and openSUSE.



To install a Debian package:  
**# dpkg -i packagename.deb**



To install an RPM package:  
**# rpm -i packagename.rpm**

While the dpkg and rpm commands look pretty straightforward, they are very difficult to use practically because of dependency. Each RPM/DEB package is always dependent on some other RPM/DEB package; if you do not have the required package in the exact version number, the install will not succeed. So in order to install one package, you have to hunt down the package it depends on, then install it. By the way, you will also have hunt down the dependent packages for the packages your original package depends on.

To work around this issue, Linux distributions have created high-level package managers which automatically download the packages and resolve all of the dependencies. The only problem with this approach is it's not standard across all distros.



On Fedora/Red Hat you can use Yellowdog Updater, Modified (YUM):

**# yum install <packagename>**

**Note:** YUM can also be installed on other distributions such as Ubuntu and openSUSE.



On Debian/Ubuntu you can use Advanced Packaging Tool (Apt):

**# apt-get install <packagename>**



On openSUSE you can use ZYpp:

**# zypper install <packageName>**

## Managing services

In Linux, a service is a crucial application (or collection of applications) that runs in the background. They handle everything from booting the system to serving webpages. You can use the command 'service' (an init script) to manage services.

To get the status of all the services installed on the system:

**# service --status-all**

To start a service:

**# service <service name> start**

To stop a service:

**# service <service name> stop**

To get the status of particular service:

**# service <service name> status**

## Running scheduled tasks

If you are doing a repetitive task on your system, it is better to automate. For example, you may want to sync files between two systems at a regular interval. Instead of doing it yourself manually, you can create a scheduled task that automatically runs at the configured intervals. In Linux (and most UNIX environments) this is achieved through cron. Cron is a time-based task scheduler.

To create a scheduled tasks using cron...

**01** Run the following command to open the current user's crontab file:

**\$ crontab -e**

If you want a task to be run using root privileges, you should use the command:

**\$ sudo crontab -e**

**02** The crontab file will then open in the default text editor.

The default text editor can be set up using the EDITOR environment variable:

**\$ export EDITOR=nano**

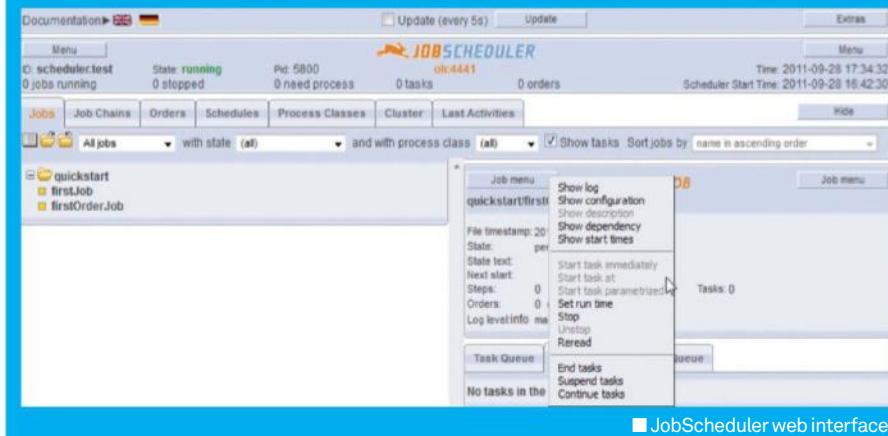
## Advanced Tip:

Cron is not the only task scheduler out there. There are a number of alternatives available. One we really like is JobScheduler. It provides the following advantages over cron:

- Provides a log file for running programs.
- The execution status of a program is checked

automatically and is reported to the administrator automatically.

- You can start jobs in a sequence that is dependent on the execution status of the jobs.
- You can use a centralised user interface to manage, configure and monitor jobs.



Crontab takes input in the following format:

minute(0-59) hour(0-23) day(1-31)  
month(1-12) weekday(0-6) command

An asterisk (\*) is used as wild card. For example, using asterisk with month will cause the task to run every month.

**03** Let's assume that you want to run /usr/bin/myludapp every day at 12.30 AM. So we will need to create the following line in it:

**29 0 \* \* \* /usr/bin/myludapp**

Here, 29 is for the 30-minute mark and 0 for 12 am because the minute, hour and weekday values start at 0. However, the day and month values start at 1 instead of 0.

## Managing backups

Backup is very crucial to any system, whether running in isolation or a networked environment. You can use rsync to create backups for your system. Rsync is a file synchronisation utility. It provides the following features which make it a perfect tool for backups:

- Differential copy: This means it will only copy the bits that have actually changed.
- On-the-fly compression: This type of compression makes the backups fast and consumes less bandwidth.

## Advanced Tip:

If you want to install and update software on multiple systems on a network, you can save a lot of bandwidth and time by creating a local software repository.

On systems which use ZYpp/YUM, you can create a local software repository using the following steps:

Firstly, mirror your desired repo to a folder, eg /var/www/ludsuserrepo/rpms.

**# sudo zypper install createrepo**

**# createrepo /var/ludsuserrepo/rpms**

At this point, all the required metadata will be added to the folder to make it a valid repository. To add this repository to the remote systems, you can use:

**# zypper addrepo -t YUM http://<host>/ludsuserrepo/rpms local\_repo**

- Security: You can use the Secure Shell protocol (SSH) to do the backups, which makes the process of backing up very secure.
- Easy to use: rsync is very easy to use, almost like the cp command but with better features.

## To do a local backup:

**# rsync -azvv <foldertobackup>**

**<destinationfolder>**

To do a remote backup over SSH (this will require OpenSSH server to be installed and started on the remote system):

**# rsync --delete -azvv -e ssh /source/folder user@remotemachine:/destination/folder**

**"If you are doing a repetitive task on your system, it is better to automate"**

Here's a breakdown of the options we've used:

- a preserves the timestamps and permissions of the files
- z compresses the data
- vv verbose output
- e sets the shell use for the transfer. Here we are specifying the SSH shell.

You can put these commands to the crontab file for regular differential backups.

## System monitoring

Monitoring is an important part of system administration. It allows you to proactively react to issues in real-time. Monitoring also gives cues on how to improve the performance of the system. The following are some of the most important command-line tools used in monitoring various components of the system...

**top:** Top provides a real-time view of the running system. It can be considered as one of the most versatile system monitoring tools out there. It displays summary information, a list of threads or processes, types of system memory, process status, CPU usage etc.

**uptime:** Uptime displays the duration for which the system has been up. It also displays how many users are currently logged on, along with the system load averages for the past 1, 5 and 15 minutes.

```
$ uptime
12:18pm up 12:22, 4 users, load average: 0.00, 0.01, 0.05
```

### Advanced Tip:

Apart from monitoring the system, you may want to monitor how individual applications are doing. Strace will help you do just that.

### Advanced Tip:

While traditionally distributions have been using the Linux init daemon to manage services, it has been replaced with modern alternatives. The most popular alternatives are systemd and upstart. Systemd is the default on Fedora/Red Hat, openSUSE, Arch Linux etc. Upstart is the default on Ubuntu, ChromeOS etc. Both of these tools provide almost the same kind of benefits, like parallel service startup and on-demand service initialisation. Both systemd and upstart are backward compatible with the init system, so init commands will work just fine.

Systemd uses the systemctl command to manage services, whereas upstart usage the initctl command for that purpose.

For example, to start a service:

```
# systemctl start foo.service
# initctl start foo.service
```

dhcppc3:/etc/init.d # top								
PID	USER	PR	NI	VIRT	RES	SHR	S %CPU %MEM	TIME+ COMMAND
7884	kunal	20	0	130m	25m	17m	S 1.0 0.8	0:10.46 konsole
3383	root	20	0	27656	4132	3380	S 0.3 0.1	0:30.11 vmtoolsd
1	root	20	0	5896	3592	2152	S 0.0 0.1	0:01.15 systemd
2	root	20	0	0	0	0	S 0.0 0.0	0:00.00 kthreadd
3	root	20	0	0	0	0	S 0.0 0.0	0:00.59 ksoftirqd/0
5	root	0	-20	0	0	0	S 0.0 0.0	0:00.00 kworker/0:0H
7	root	0	-20	0	0	0	S 0.0 0.0	0:00.00 kworker/u:0H
8	root	rt	0	0	0	0	S 0.0 0.0	0:00.00 migration/0
9	root	20	0	0	0	0	S 0.0 0.0	0:00.00 migration/0
10	root	20	0	0	0	0	S 0.0 0.0	0:00.82 rcu_sched
11	root	rt	0	0	0	0	S 0.0 0.0	0:00.19 watchdog/0
12	root	rt	0	0	0	0	S 0.0 0.0	0:00.16 watchdog/1
13	root	20	0	0	0	0	S 0.0 0.0	0:00.55 ksoftirqd/1
14	root	rt	0	0	0	0	S 0.0 0.0	0:00.52 migration/1
16	root	0	-20	0	0	0	S 0.0 0.0	0:00.00 kworker/1:0H
17	root	0	-20	0	0	0	S 0.0 0.0	0:00.00 cpuset
18	root	0	-20	0	0	0	S 0.0 0.0	0:00.00 khelper
19	root	20	0	0	0	0	S 0.0 0.0	0:00.00 kdevtmpfs
20	root	0	-20	0	0	0	S 0.0 0.0	0:00.00 netns
21	root	20	0	0	0	0	S 0.0 0.0	0:00.00 bdi-default
22	root	0	-20	0	0	0	S 0.0 0.0	0:00.00 kintegrityd
23	root	0	-20	0	0	0	S 0.0 0.0	0:00.00 kblockd
24	root	0	-20	0	0	0	S 0.0 0.0	0:00.00 md
27	root	20	0	0	0	0	S 0.0 0.0	0:00.00 khungtaskd
28	root	20	0	0	0	0	S 0.0 0.0	0:00.00 kswapd0

■ Output from the top command

**sysstat performance tools:** Most distributions do not include sysstat by default, but you can easily install it using your distribution's package manager. Systat includes the following tools:

- **iostat:** reports CPU utilisation and disk I/O statistics;
- **mpstat:** reports global and per-processor statistics;
- **pidstat:** reports statistics for Linux tasks (processes);
- **nfsiostat:** reports I/O statistics for network file systems;
- **cifsiostat:** reports I/O statistics for CIFS file systems

• **sar:** collects and reports system activity information;

These tools are very helpful in monitoring I/O across the whole system.

**\$ iostat**

**avg-cpu:**

%user	%nice	%system	%iowait	%steal	%idle
0.85	0.05	2.51	6.14	0.00	90.44

**Device:**

	tps	kB_ read/s	kB_ wrtn/s	kB_ read	kB_ wrtn
sda	35.79	613.38	38.54	519671	32648
fdo	0.00	0.01	0.00	8	0

**pmap:** pmap reports a memory map of

**\$ sudo mpstat**

12:47:37 PM CPU	%usr	%nice	%sys	%iowait	%irq	%soft	%steal	%guest	%idle
12:47:37 PM all	0.82	0.05	2.29	5.81	0.00	0.10	0.00	0.00	90.92

**\$ pidstat**

12:48:41 PM	PID	%usr	%system	%guest	%CPU	CPU	Command
12:48:41 PM	1	0.01	0.17	0.00	0.18	0	init
12:48:41 PM	2	0.00	0.00	0.00	0.00	0	kthreadd

**\$ pmap -d 3275**

Address	Kbytes	Mode	Offset	Device	Mapping
0000000000400000	900	r-x--	0000000000000000	008:00001	bash
00000000006e0000	4	r----	000000000000e000	008:00001	bash
000000000025fc00	2076	rwx--	0000000000000000	000:00000	[anon]
00007f0e5f20b000	2044	-----	000000000000c000	008:00001	libnss_files-2.15.so

mapped: 26960K writeable/private: 2356K shared: 28K

**Advanced Tip:**

If you are looking for a more advanced solution for backup you can use Bacula ([www.bacula.org](http://www.bacula.org)). It is a fully fledged open source network backup solution. It also has its own ecosystem of add-ons which includes everything from specialised monitors and report builders to even a Bacula-specific file system (BaculaFS).

a process. It is very helpful in detecting memory bottlenecks.

```
$ pmap -d 3275
```

**iptraf:** iptraf is a TCP/UDP network monitoring utility. It has a nice ncurses-based user interface which liberates users from having to remember any command-line switches.

**strace:** strace intercepts and records the system calls which are called by a process and the signals which are received by a process. The name of each system call, its arguments and its return value are printed on standard error or to the file specified with the -o option. Strace is a useful diagnostic, instructional and debugging tool. It is particularly good for solving problems with programs for which the source is not readily available, since they do not need to be recompiled in order to trace them.

```
$ strace wget www.rarlab.com/rar/winrar-x64-420.exe
execve("/usr/bin/wget", ["wget", "http://www.rarlab.com/rar/winrar..."], /* 43 vars */) = 0
brk(0) = 0x2463000
access("/etc/ld.so.nohwcap", F_OK) = -1
ENOENT (No such file or directory)
munmap(0x7f259cb5f000, 4096) = 0
stat("/home/kunal/.wgetrc", 0x7fff01fb9010) = -1 ENOENT (No such file or directory)
write(2, "Connecting to www.rarlab.com (www.rarlab.com)|188.138.1.135:80...") = 67
socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 3
connect(3, {sa_family=AF_INET, sin_port=htons(80), sin_addr=inet_addr("188.138.1.135")}, 16) = 0
write(2, "connected.\n", 11connected.
```

As you can see in the above example, we are using strace to obtain detailed information about everything wget is doing since we have issued the command. This includes the files it has opened, network connections it has made and so on.

**Distribution-specific GUI administration/monitoring tools**

While command-line and web-based administration are very powerful, GUI administration tools are easier and simpler to use. In this section we will look at some of the best GUI administration tool available on modern Linux distributions.

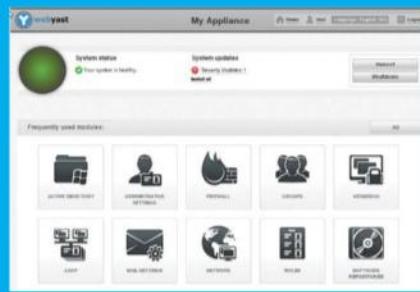
**YaST2:**

YaST (Yet another Setup Tool) is the installation and configuration tool for SUSE Linux distributions. YaST was one of the first to introduce a centralised configuration tool rather than having many single application utilities. YaST is an all-in-one solution which allows users to configure every aspect of a system, including managing packages, printers, sound system, kernel, partitioning, users etc. Configuration options are categorised under Software, Hardware, System, Network Devices, Network Services, Security and Users, Support, and Miscellaneous. All the configuration utilities provide an easy-to-use wizard-based interface. All YaST modules contain a dynamic help button for users who want more information on the configuration they are performing.

One of the key features that set YaST apart is its curses-based easy-to-use interface. It is very helpful for people who want to use all the power of YaST in text mode. YaST also includes a Ruby-based web interface called WebYaST, which provides all the features of YaST over the web.

YaST2 uses a modular architecture and additional modules can be developed using the YaST2 SDK.

YaST2 is included in all openSUSE Linux distros (as well as the commercial SUSE ones).

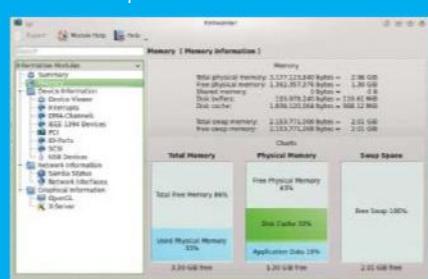


■ YaST2 curses-based text interface

**KInfoCenter**

KInfoCenter is KDE utility which provides hardware and graphics information. Most of this information is directly polled from the Linux kernel's /proc file system.

KInfoCenter is included in the standard KDE Software Compilation.

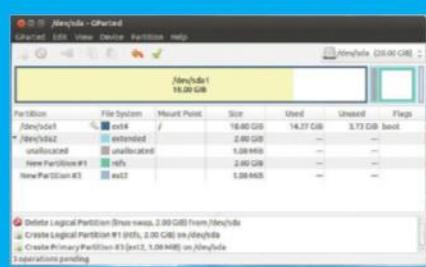


■ KInfoCenter

**GParted**

Parted is an excellent disk partitioning tool, but when it's not used carefully it can destroy data. That's where GParted comes in. It is an excellent GUI interface to Parted. It is easy to use and looks almost like the insanely popular Windows software Partition Magic. GParted uses a workflow-based approach to disk partitioning. Modifications are not applied automatically – instead, the user gets a chance to review the changes and can apply them only if he or she is comfortable with it.

GParted is a tool that is included with the GNOME Software distribution.



■ GParted Partition Manager

**iPTraf monitoring TCP Connections**

		Packets	Bytes In	Bytes Out
192.168.138.128:60925	=	25	1619	--A-
91.189.89.89:80	=	24	28952	-PA-
84.45.8.18:80	=	80	82356	CLOSED
192.168.138.128:55288	=	53	7588	CLOSED
50.18.169.106:80	=	7	4313	-PA-
192.168.138.128:36339	=	8	925	--A-
192.168.138.128:55279	=	77	9346	CLOSED
84.45.8.18:80	=	121	134509	CLOSED
192.168.138.128:59913	=	21	8742	--A-
74.125.236.80:80	=	25	3016	-PA-
192.168.138.128:55282	=	72	7454	CLOSED
84.45.8.18:80	=	115	137491	CLOSED

# Using a system administration configuration suite

In this section, instead of focusing on individual tools we will look at a solution which gives a full set of tools for system administration in one place.

## System admin using Webmin

Webmin is a web-based system administration tool for a variety of UNIX-like systems. Webmin also has a vibrant ecosystem of modules around it. These modules extend the feature of Webmin to cover new applications and services.

Webmin is available for all the major Linux distributions. You can download it from: [www.webmin.com/download.html](http://www.webmin.com/download.html)

The easiest way to install it is from your distribution's package manager. If it is not

## Selecting the user interface for system administration

### Command-line tools

#### Positive:

- ✓ They are easily accessible from within the system or remotely (using SSH or telnet).
- ✓ They can be also used on a system with a low amount resources and are very handy in recovering a system which has only a command-line interface available.
- ✓ Command-line tools are easy to automate using scripts.

#### Negative:

- ✗ They are complex and more difficult to use than their web or GUI counterparts.

### GUI tools

#### Positive:

- ✓ They are very easy to use and are often included with the distribution you are using.
- ✓ When designed properly, they give access to most options and provide automatic help and documentation right from the user interface.

#### Negative:

- ✗ Difficult to access from a remote system.
- ✗ Hard to automate.
- ✗ For each Linux distribution you may need to use different set of tools.

### Web tools:

#### Positive:

- ✓ Easy to use.
- ✓ Can be accessed remotely.

#### Negative:

- ✗ Hard to set up.
- ✗ Security hole when not configured properly.

available in the package manager, you can download a DEB or RPM package from the Webmin site downloads page and install it directly on your system. After installing Webmin, it is available at <https://localhost:10000>. Here you'll need to log in with the root credentials. If you are using Ubuntu, then you will need to create a root password. You can create a root password using the following command:

```
$ sudo su  
# passwd
```

### 1. Managing services

Expand System on the navigation bar, then click on Bootup and Shutdown. Here Webmin will list the type of boot system in use and all the services. It will also show if the service will start at boot and its current status.

Clicking on any service will open the service script. You can make changes to the service



■ Webmin default page

script and set its boot-time status. You can also start or stop the service from here.

### 2. Managing processes

Expand System, click on Running Processes. Here you'll see the Running processes list with process ID, Owner, Process Start Time and Command. Using the Display option, you can view user, memory and CPU usage as well.

Click on the process ID that you want to view/edit. On the process information screen, you can see the command, process information, owner and size among other details. You can use this screen to trace the process, see its open files and connections, or kill the process.

### 3. Configuring Apache web server

Traditionally, configuring Apache web server means editing the httpd.conf file. Webmin makes it very easy to configure Apache web server by providing a nice GUI interface to the Apache configuration files. To configure Apache web server, expand Servers on the navigation bar, then click on Apache Web Server. By default it will open in the Virtual Hosts. If you want to change Global Configuration, you can click on the Global Configuration tab. Click on a

Running Processes			
ID	Owner	Started	Command
1	root	12:32	/sbin/init
459	root	12:33	upstart-udev-bridge --daemon
464	root	12:33	/sbin/uddev --daemon
816	root	12:33	/sbin/uddev --daemon
917	root	12:33	/sbin/uddev --daemon
498	root	12:33	/usr/sbin/sshd -D
584	syslog	12:33	rsyslogd -c5
699	messagebus	12:33	dbus-daemon --system --fork
721	root	12:33	/usr/sbin/bluetoothd
731	avahi	12:33	avahi-daemon: running [KunalUbuntu.local]

■ Running processes list

Process Information	
Command	nautilus -m
ID	2858
Owner	kunal
Size	1358116 kB
Nice level	0 (Default) Change
IO scheduling class	Default
Real group	kunal
Group	kunal
Started	12:34
Send Signal	HUP
Terminate	Kill
Suspend	Resume
Trace Process	Files and Connections

■ Detailed process information

The screenshot shows the Apache Webserver configuration interface. It includes tabs for 'Global configuration', 'Existing virtual hosts' (selected), and 'Create virtual host'. Under 'Existing virtual hosts', there are three entries:

- Default Server:** Handles the name-based server on address \*. Address Any, Server Name Automatic, Document Root Automatic.
- Virtual Server:** Handles the name-based server on address \*. Address Any, Server Name Automatic, Document Root /var/www.
- Virtual Server:** Handles the name-based server on address \*. Address Any, Server Name Automatic, Document Root /var/www.

Buttons at the bottom include 'Delete Selected Servers'.

#### Apache web server configuration

Virtual Host to modify it. Here you can configure options related to the virtual server, such as directory, MIME types, port, server name etc. Creation of a new virtual server configuration is also very easy: you can click on the Create Virtual Host tab to create a new Virtual Server Configuration.

#### 4. Special features

Apart from system configuration features, Webmin also provides a few utilities which are excellent for new system administrators...

**File Manager:** Webmin comes with a built-in fully featured file manager. It is excellent for admins who want to make changes to the file system on the server. File Manager also comes with a handy editor which is excellent for making changes to configuration files. File Manager can be accessed via Others>File Manager. Note that File Manager requires a Java plug-in to be enabled on the browser side.

**Built-in terminal:** Most system admins would really appreciate having shell access to the server. But it is not always available everywhere. Webmin includes a nice little utility called Text Login which provides shell access to the server. It can be run on any browser and does not depend on Java. To access the shell, click Others>Text Login. Keep in mind that some systems do not allow root login from a remote shell. In this case you will need to use a regular user for login and then use su for performing administrative tasks.

**Webmin modules:** Webmin has a thriving community of module makers. You can use these modules to add features to Webmin. Installing Webmin modules is very easy. Go to

The screenshot shows the 'Virtual Server Options' configuration interface. It includes tabs for 'Module Index' (selected), 'Virtual Server Options', and 'Per-Directory Options'. Under 'Virtual Server Options', there are several icons for managing virtual servers. Under 'Per-Directory Options', there is a 'Create Path Directory' dialog with fields for 'Type' (set to 'Directory'), 'Regexp' (set to 'Exact match'), and 'Path' (set to '/'). Under 'Virtual Server Details', there are fields for 'Address', 'Port', 'Document Root', and 'Server Name'.

#### Conclusion

There are some pretty fat books written about Linux system administration. This article was not an attempt to create an all-in-one guide, but a humble attempt to cover important things and get you excited to explore more.

The screenshot shows the 'Webmin Modules' interface. It includes tabs for 'Module Index' (selected), 'Webmin Modules' (selected), and 'Install Module'. Under 'Install Module', there is a form for selecting the source of the module: 'From local file', 'From uploaded file', 'From ftp or http URL', 'Standard module from www.webmin.com', or 'Third party module from' (with a text input field for the URL). Below this is a section for 'Ignore dependencies?' with radio buttons for 'Yes' and 'No'. Under 'Grant access to', there are radio buttons for 'Grant access only to users and groups : root' and 'Grant access to all Webmin users'. A 'Select Third Party Module - Google Chrome' window is open, showing a list of available modules:

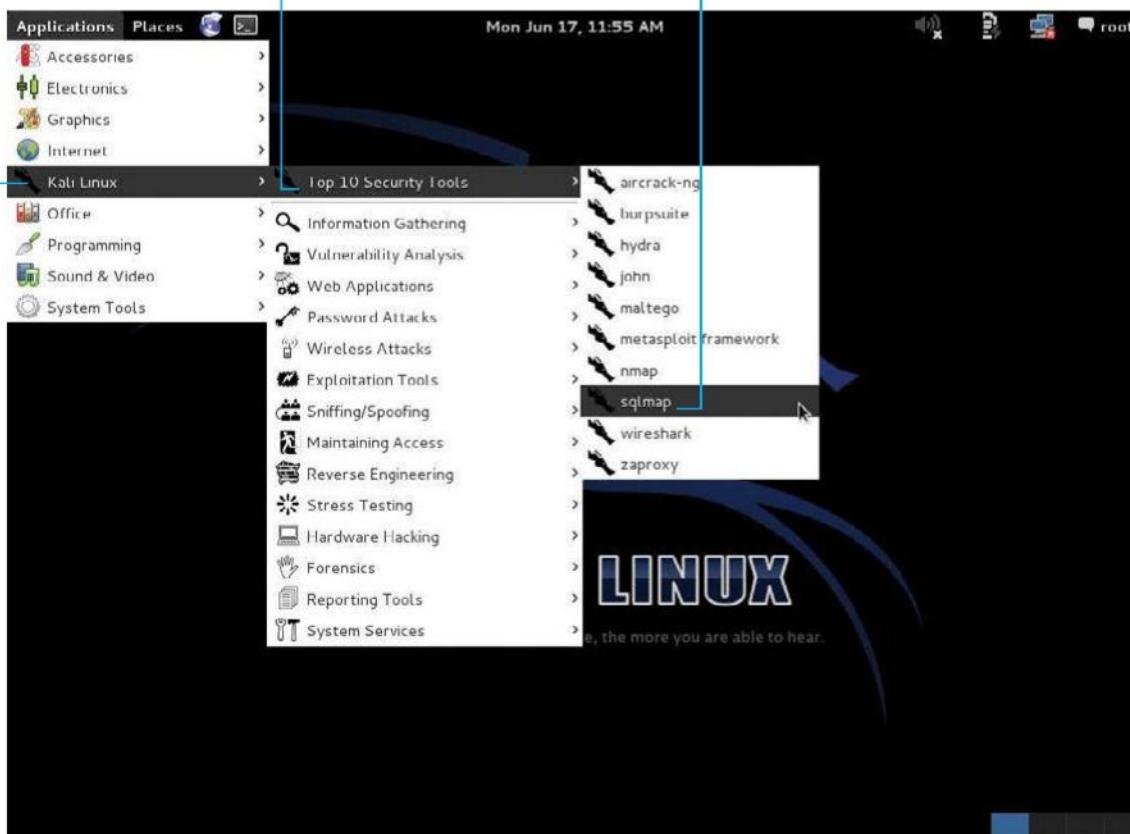
Name	Version	Description
b3ddns	1.0.4	module, but only supports BIND version 9 and above. Includes new named.conf options, date-based SOA support, IPv4 and IPv6 support, DNAME support, etc.
Bacula Restore	1.6	Restore backups created using the open-source Bacula package. Supports browsing through the hierarchy of backed-up files in the database to select those to be restored. Now also supports a PostgreSQL Bacula database.
Dacula Restore	1.3	Restore backups created using the open-source Dacula package. Supports browsing through the hierarchy of backed-up files in the database to select those to be restored. Now also supports PostgreSQL and SQLite Bacula databases.
bannerfilter	0.3	Configure the bannerfilter plugin for the Squid Web Proxy Cache.
Batch Apache	0.2.2	Automates the creation of multiple Apache websites.
Batch DNS	0.1.2	Automates the creation of multiple DNS domains.
BB	0.05	Big Brother Configuration
BitDefender Remote Admin	1.5.6	BitDefender Remote Admin allows system administrators to manage BitDefender Antivirus and Antispam for Linux remotely via webmin.

#### Webmin modules

Kali starts up with a top-level menu entry. Almost all of the tools available will be listed here, making it easy to start testing your security

The top ten applications that are used most often have their own menu entry. This saves you having to hunt in the submenus

Each application has an entry in the menu. If it is a console-based application, it opens in a new terminal with a listing of the options for that tool



# Test your network's security

One of the best ways to test your security is to try to tear it apart, and you can do just that with Kali Linux...

## Resources

Kali Linux: [www.kali.org](http://www.kali.org)

Metasploit: [www.metasploit.com](http://www.metasploit.com)

**Security is something that everyone needs to be aware of and something that everyone needs to deal with.** While you can go out and collect a number of tools and utilities to help you out, there is an easier path. There are several Linux distributions out there that provide an entire suite of tools to fit your security needs. One of the more popular ones is Kali Linux (originally BackTrack). There are other ones, like BackBox or Lightweight Portable Security, which may fit specific needs better. You can run these off of a bootable DVD

or USB drive, allowing you to run forensics on a compromised machine. Alternatively, you could install it on a box and set it up on your network for a more permanent security solution.

In this tutorial, we'll use Kali Linux to go through one possible set of steps to analyse and test your local security. We will only be able to cover a subset of all of the tools available in Kali Linux, but you will learn some basic techniques to monitor your systems and to test your defences of them.



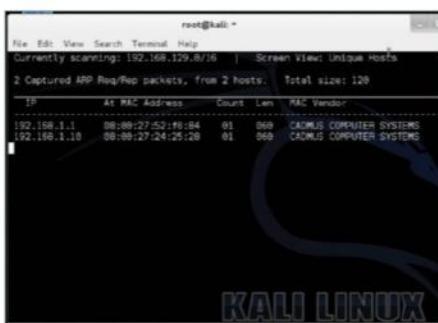
## 01 Download and install

The first step is to get a copy of Kali Linux to work with. The main download page provides downloads in several formats and for several different architectures. The usual thing to do is to download an ISO and either burn it to a CD or create a bootable USB drive.



## 02 Hardware detection

One cool extra that Kali Linux provides is the ability to take a look at your hardware before booting up. It is always a good idea to get a lay-of-the-land look at the hardware you want to investigate. This is a boot option when you start up Kali.



## 03 Netdiscover

One of the first things to do is to find out who, or what, is on your network. Netdiscover gives you a tool to do IP address mapping on your network. This is especially useful on Wi-Fi networks that aren't using DHCP.

## 04 Tcpflow

Once you have a list of hosts, then you will probably want to look at what kind of communication is happening. Tcpflow will

```

root@kali:~#
File Edit View Search Terminal Help
-h: max number of bytes per flow to save
-c: console print only (don't create files)
-C: console print only, but without the display of source/dest header
-d: debug level; default is 1
-e: output each flow in alternating colors
-f: maximum number of file descriptors to use
-h: print this help message
-i: network interface on which to listen
    (type "ifconfig -a" for a list of interfaces)
-p: don't use promiscuous mode
-r: read packets from tcpdump output file
-s: strip non-printable characters (change to '.')
-v: verbose operation equivalent to -d 10
expression: tcpdump-like filtering expression

See the man page for additional information.

root@kali:~# tcpflow -i eth1 -c -v
tcpflow[5104]: tcpflow version 0.21 by Jeremy Elson <jelson@circlemud.org>
tcpflow[5104]: looking for handler for datalink type 1 for interface eth1
tcpflow[5104]: found max FDs to be 16 using OPEN_MAX
tcpflow[5104]: listening on eth1
The quieter you become, the more you are able to hear

```

monitor the traffic occurring on your network and construct conversations you can analyse to see what your network is being used for.

```

root@kali:~#
File Edit View Search Terminal Help
InTrace 1.5 -- R: 192.168.1.10/52426 (6) L: 192.168.1.11/80
Payload Size: 1 bytes, Seq: 0x1c867a0b, Ack: 0x00000000
Status: Press ENTER

# [src addr] [icmp src addr] [pkt type]
1. [192.168.1.10] [ ... ] [TCP_RST]

```

## 05 Intrace

Once you know what conversations are occurring on your network, you may be interested in finding out what routes those conversations are taking. Intrace gives you a traceroute-like listing of packet paths by looking at the TCP packets flowing on your network.

IP	At MAC Address	Count	Len	MAC Vendor
192.168.1.1	08:00:27:52:16:84	61	668	CADMUS COMPUTER SYSTEMS
192.168.1.10	08:00:27:24:25:28	61	668	CADMUS COMPUTER SYSTEMS

## 06 Zenmap

After identifying the hosts on your network, you will probably need to see what ports are open on them, and what OS is running there. The go-to application for this is Nmap. The usual GUI front-end used for Nmap is Zenmap.

```

root@kali:~#
File Edit View Search Terminal Help
Sqlninja rel. 0.2.4-73
Copyright (c) 2009-2010 iceisurfer <x86@northernfortress.net>
Usage: sqlninja [options] target
  -m module : Required. Available modules are:
    ftest - test whether the injection is working
    fingerprint - fingerprints user, xp_cmdshell and more
    http - attempt to exploit http services
    reverse - attempt to gain a reverse shell
    escalation - attempt to gain a higher server role
    recursive - try to recreate xp_cmdshell
    upload - upload a .scr file
    cmdshell - start a direct shell
    http - attempt to exploit an outbound port
    reverse - start a reverse shell
    d��nshell - attempt a direct tunnel
    lircmdshell - start a reverse IIRC shell
    zigbee - attempt to exploit zigbee
    msfexploit - wrapper to Metasploit stages
    -f file : configuration file (default: sqlninja.conf)
    -p password : as password
    -w wordlist : wordlist to use in bruteforce attack (dictionary method)
    -g generate debug script and exit (only valid in upload mode)
    -v : verbose output
    -d mode : activate debug
        1 - print each injected command
        2 - print each HTTP request
        3 - print each raw HTTP responses
    all - all of the above
... see sqlninja.howto.html for details
root@kali:~# 

```

## 07 Sqlninja

Now we need to start poking at security. Microsoft is always a punching bag when it comes to security, and SQL Server is no exception. Most corporate networks use Microsoft software, so you need to test how they are configured. Sqlninja is the tool to beat on SQL Server, using techniques like SQL injection.

## 08 Acccheck

Another service that can prove to be a weak point in your systems is SMB, or Samba file sharing. The utility acccheck can be used to run a dictionary attack on account passwords, trying to break through Windows authorisation.

**"We need to start poking at security"**

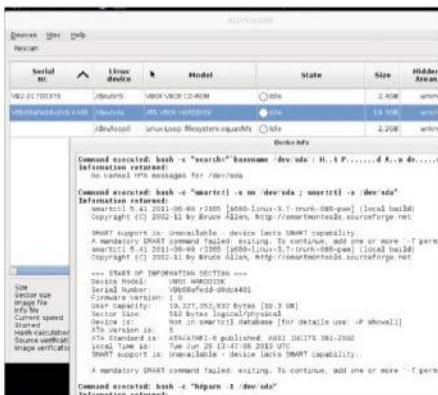


## 09 Forensics mode

**09** If you do find a machine that you think may have been compromised, you want to be careful when you try to investigate it. Kali Linux provides a forensics mode on bootup that simply boots up and leaves all local drives unmounted and untouched. That way, you can run tests without changing the state of the system.

## 10 Offline password cracking

**10** One of the things you will want to investigate is if the machine has been compromised due to weak password selections. There are several tools that can be used to try to crack password hashes. Most of these, like John the Ripper, use dictionary attacks to dig out passwords.



11 Guymager

In some cases, the machine in question may be too important to leave offline. In these cases, the only option is to make an image of the drive to investigate later before rebuilding. Guymager is one of the tools available to make images for this purpose.

**“Once you have your network secured, that is only the beginning”**

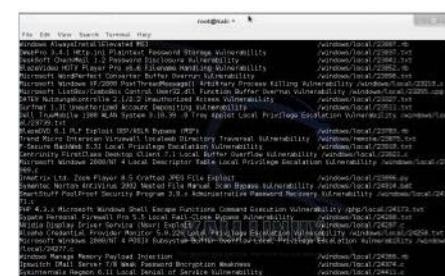


12 Chkrootkit

**12** One of the things you will need to look for during an investigation is whether a rootkit has been installed, providing a back entrance to the bad guys. One of the tools you can use to do this is chkrootkit. This utility looks for evidence of common rootkits used for taking over machines.

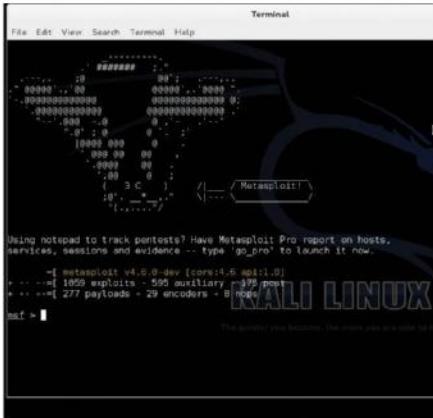
## 13 Social engineering

**13 Social Engineering** One aspect of security that gets neglected is the social aspect. All of the security in the world won't help if your users aren't computing safely. Kali Linux provides a social engineering toolkit that you can use to do things like trying out spear-phishing attacks.



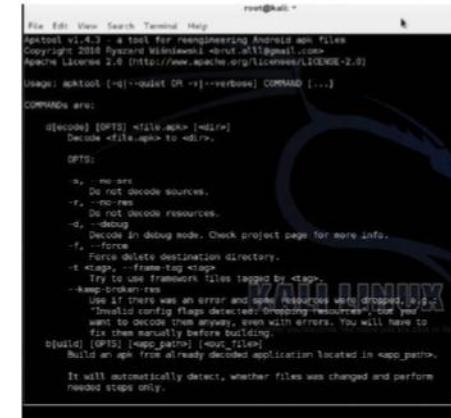
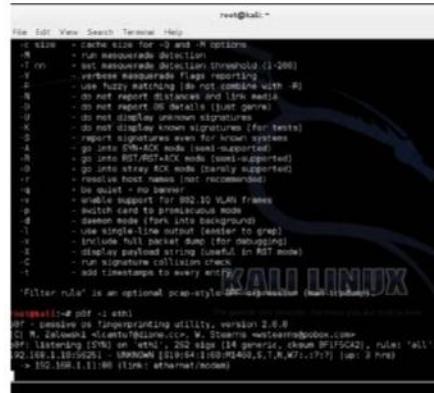
14 Exploit databases

**14** Along with testing the social aspect, you also need to test the security of the machines to find any holes. You do this by trying known exploits. Luckily, there is a database full of known exploits online.



## 15 Metasploit

The usual tool used to test a system is Metasploit, which provides a full framework for putting together complete attack vectors. These include intrusions, compromises and channels to allow for remote access of a compromised machine. Within Kali Linux, there are menu items that allow you to start up the Metasploit server. There's also an entry to grab a dump of diagnostic logs, in case you run into issues. Metasploit runs in a client-server model, so once you start up the server, you will need to connect with a client in order to try some exploits against the machines that you are responsible for.

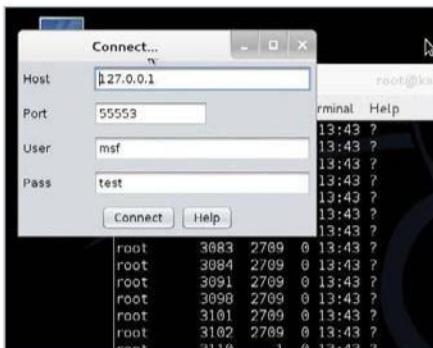


tool passively monitors a network to see what machines exist and what OS they run, without letting them know that you are listening.



## 18 Hardware exploits

One set of tools that Kali provides that is unique is the ability to test other hardware. There are tools to poke into Android devices, Bluetooth protocols and Arduino systems.

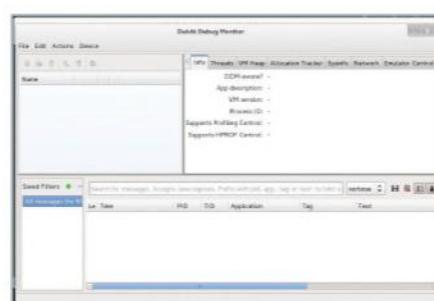


## 16 Armitage

One of the graphical interfaces available to you is Armitage. If you have already started Metasploit, then you can tell Armitage to connect to this already-running server. Otherwise, Armitage can start up a new Metasploit server for you to play with.

## 17 p0f

Once you have your network secured, that is only the beginning. You need to keep up with what is happening on your system. The p0f



## 19 DDMS

DDMS is a debugging monitor that gives you low-level access and control of Android machines. You simply need to plug your device into a USB port, start up DDMS and check out what is happening on the device. You do need to install an SDK for a specific version before starting.

## 20 Android exploits – apktool

Once you have your Android device attached, you can run various exploits to get root access. These vary, based on what kind of hardware your Android is running on. One type of exploit may need apktool, in order to open and edit the APK files on your Android device.

## 21 Bluetooth

You also have another possible security hole. The Bluetooth protocol is used for mice, keyboards and other bits of hardware. But security was never really thought of in any major sense. Kali Linux provides several tools to look at the Bluetooth signals travelling around.



## 22 Install on ARM

Support from the Kali developers has provided for an ARM architecture version. You can find it on the main download page. There are even instructions on how to install it on a Galaxy Note 10.1 device, including an installation image.

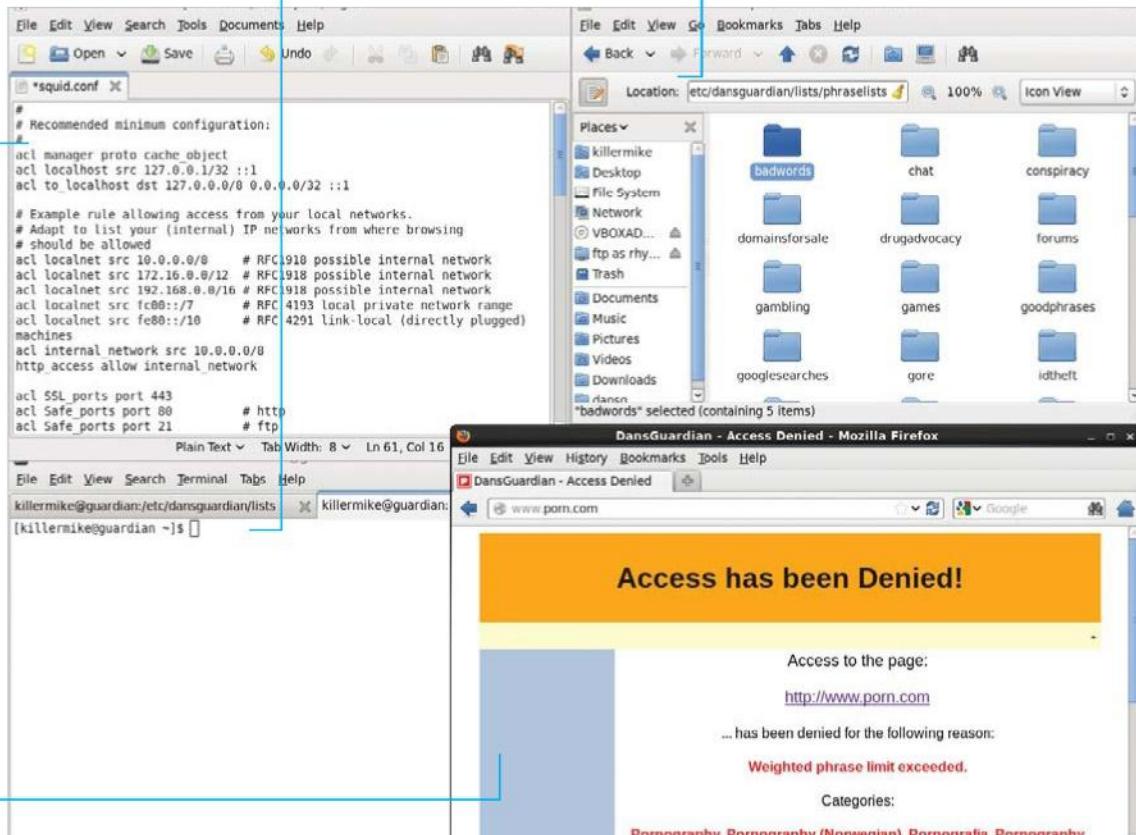
## 23 Conclusion

Hopefully, if you follow these steps, you can start to get a handle on the security needs for your system. This is only a start, though. There are lots more tools available in Kali Linux than we covered here, so don't be afraid to check out what else is available.

Most of the configuration of the components is carried out using text files

We'll be spending a bit of time at the command line for this one

We're basing this project around a fresh installation of CentOS 6, but most of it can be applied to other distros



The finished result is a system that filters out the type of material that you tell it to, in an intelligent way

# Protect your network

Build a gateway server that can intelligently filter content and block access to certain websites from certain PCs

**This is a project to create a gateway PC that allows you to filter internet traffic.** We're going to use CentOS as the base of our system and the web filter DansGuardian will carry out the filtering for us.

Filtering the internet has never been more topical, and running DansGuardian puts that power into the hands of the administrator. Basic filtering software blocks individual pages, but DansGuardian is adaptive and analyses the content of pages on the fly. Even better,

DansGuardian carries out a sophisticated analysis of the content that uses weighted trigger phrases. This means that a single instance of a banned word might not block the page that the user is attempting to access.

The gateway PC sits between your broadband internet connection and the rest of your network and is capable of assigning connection details to client PCs using DHCP. These computers will lack a direct connection to the internet until you configure them to use our proxy setup.

## Resources

- Server machine
- Two Ethernet adaptors
- Firefox web browser



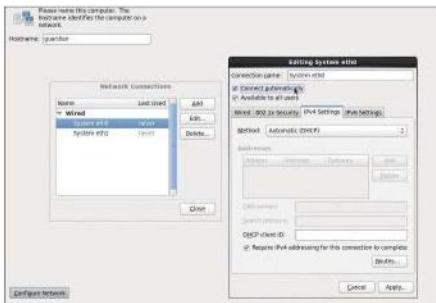
## 01 Set up server

Our example network layout revolves around a single server PC with two network adaptors – one connects to the internet (via router or modem) and the other to the rest of the network (via switch or hub). A Wi-Fi connection to outgoing connection is acceptable if it'll meet the bandwidth requirements of your network.



## 02 Install CentOS

Download the latest CentOS DVD image from [www.centos.org](http://www.centos.org). This installation is fairly standard until you get to the networking page. Give the computer a hostname, such as guardian, and then click on Configure Network.



## 03 Set up the adaptors

Click on a network adaptor, then on Edit... to edit the settings for each one in turn. Select the first adaptor and check 'Connect automatically'. Now select Method: Manual in the IPv4 tab. Give the first adaptor an address of 10.0.2.100, a netmask of 255.255.255.0 and a gateway corresponding to the IP address of your router. Give the second adaptor an IP address of 10.0.3.100. Accept the changes, then select Desktop installation profile and wait for the installation to complete. Upon reboot, create a basic user when prompted and then log in.



## 04 Become root

For most of this tutorial, you'll need to run as root. In CentOS, you can become root by typing `su` and then inputting the root password. For the bits that don't need root access, consider hitting `Ctrl+T` in the terminal window to create a tab with normal user access.



## 05 Install the repository

Visit the CentOS RPMForge page (Google for it or go to [tinyurl.com/4gjcxz](http://tinyurl.com/4gjcxz)) and follow the instructions there to download the `rpmforge-release` package. Install DAG's GPG key as instructed. Now install the package with `rpm -i [name of package].rpm`. Carry out a `yum update` to update the system.



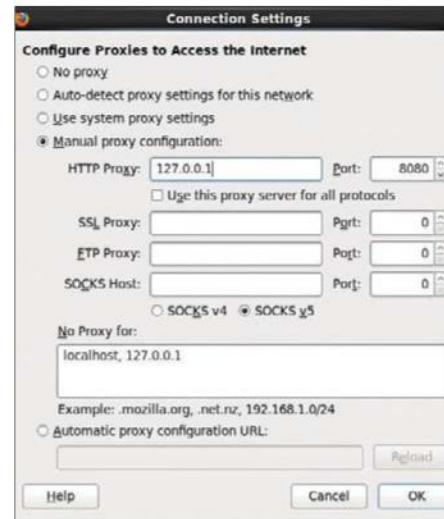
## 06 Install DansGuardian and Squid

DansGuardian and web cache Squid work in tandem with each other. Install them both by issuing the command `yum install dansguardian squid`.



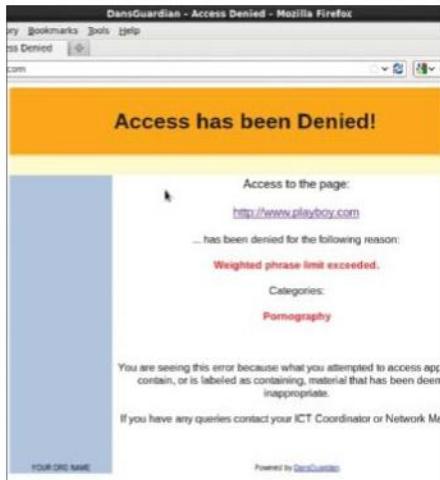
## 07 Start DansGuardian and Squid

We're going to use the `service` command to control all services. Start DansGuardian with `service dansguardian start` and then start Squid with `service squid start`. Check the output of both commands for errors.



## 08 Test the proxy

Odds are, Squid and DansGuardian are working acceptably well with the default settings. To test this, we're going to select DansGuardian as the default proxy. Launch Firefox and go to `Edit>Preferences>Advanced>Network`. Now select the `Settings...` button. In the `Connection Settings` dialog, select 'Manual proxy configuration'. In the `HTTP Proxy` box, insert `127.0.0.1` with a port of `8080`.

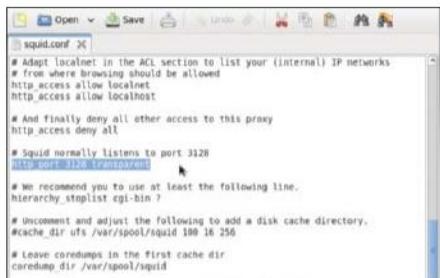


## 09 Test the proxy

Accept the changes you have just made and type `wikipedia.com` into the URL bar. If everything's working, the page should display as normal. If you're in a public place, choose a fairly tame site that should be blocked for testing. You should now see DansGuardian's default block page.

## 10 Configure Squid

Type `sudo gedit /etc/squid/squid.conf` & to open the Squid configuration. Add the lines `acl internal_network src 10.0.0.0/8` and `http_access allow internal_network`. In other words, process requests from machines with IP addresses that begin 10.x.x.x, which is our LAN. Add the line `visible_hostname guardian`. Type `service squid restart` to restart Squid.



## 11 Add DHCPD

Type `yum install dnsmasq`. Machines connected to the eth1 subnet need to be

```
# Override the default route supplied by dnsmasq, which assumes the
# router is the same machine as the one running dnsmasq.
#dhcp-option=3,1.2.3.4
#
# Do the same thing, but using the option name
#dhcp-option=option:router,1.2.3.4
#
# Override the default route supplied by dnsmasq and send no default
# route at all. Note that this only works for the options sent by
# default (1, 3, 6, 12, 28) the same line will send a zero-length option
# for all other option numbers.
#dhcp-option=3
#
# Set the NTP time server addresses to 192.168.0.4 and 10.10.0.5
#dhcp-option=option:ntp-server,192.168.0.4,10.10.0.5
#
# Set the NTP time server address to be the same machine as

```

assigned an IP address. Edit `/etc/dnsmasq.conf`. Add the lines (without comments)...

```
interface=eth1 #Only activate on the LAN
dhcp-option=eth1,3,10.0.2.100 #Specify the
gateway
dhcp-range=eth1,10.0.3.10,10.0.3.200,255.255
.255.0,24h # Assign IP addresses 10.0.3.10
- 10.0.3.200.
```



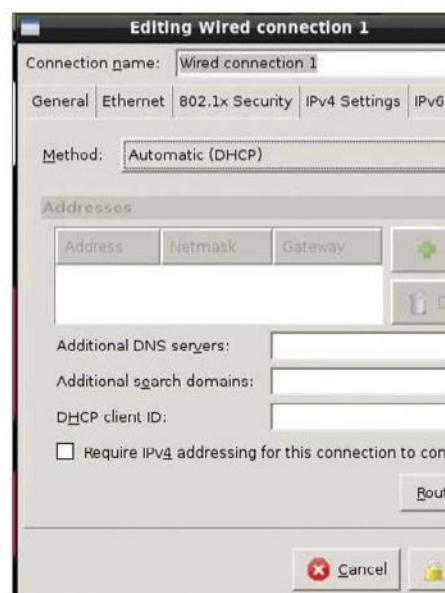
## 12 Configure services and restart

Type `chkconfig --add <service name>` followed by `chkconfig <service name> on`. Do this for the following services: dnsmasq, dansguardian, squid. Now restart the machine.

**“Keep this list a secret and then assign a static IP to machines that require unfiltered access”**

## 13 Configure the clients

Connect a machine to your LAN and make sure DHCP is selected on the client. The machines on the LAN should be assigned an IP address on startup – confirm by typing `ifconfig` into a terminal. In Firefox, set up the proxy as before, but add 10.0.3.100 as the IP address and check 'Use this proxy server for all protocols'.



## 14 Configure DansGuardian behaviour

Most of the files that control the filtering behaviour of DansGuardian reside within `/etc/dansguardian/lists/` and you can guess many of

```
killer@guardian:~$ ls /etc/dansguardian/lists/
authplugins      contentscanners      filtergroupslist
bannedextensionlist downloadmanagers   greysitelist
bannediplist    exceptionextensionlist greyurllist
bannedmediatypeplist exceptionfilelistlist headerregexpelist
bannedphraseplist exceptionfileurllist  logregexpurllist
bannedregexheaderlist exceptioniplist   logurllist
bannedregexurllist exceptionimmetypelist phraselists
bannedsiteplist  exceptionphraselist  pics
blacklists       exceptionregeurllist  urllist
contentregeplist exceptionsiteplist   weightedphraselist
exceptionurllist
```

their functions from the title. When you make a change to these files, restart DansGuardian with `service dansguardian restart`.

## 15 Add IP exceptions

`/etc/dansguardian/lists/exceptioniplist` contains a list of client machines that will be not be subjected to any content filtering. Keep this list a secret and then assign a static IP to machines that require unfiltered access.

```
# This is not the IP of web servers
# you don't want to filter.

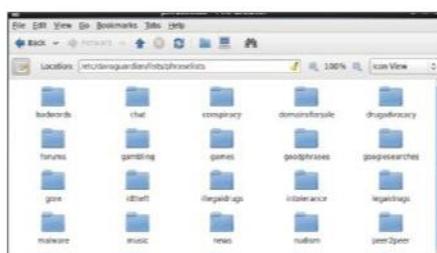
#192.168.0.1
#192.168.0.2
#192.168.42.2

# Ranges and subnets can also be used,
# e.g.
# 10.0.0.1-10.0.0.3
# 10.0.0.0/24

# Hostnames can also be used, provided
# you cater for reverse DNS lookups
# on your LAN and enable the
# "reverseclientiplookups" option in
# dansguardian.conf
```

## 16 Add to banned phrases

For ease of management, `bannedphraselist` includes lists from within the `/phraselist` subdirectory. However, you can add phrases in this top-level configuration file, and the format is explained in the file itself. Usefully, it's easy to specify combinations of words that trigger the blocker.



The screenshot shows a browser window with the URL `URL.BlackList.com`. The page displays a download section for a file named `bigblacklist.tar.gz` with details: Date Modified: 2013-09-09, Size: 21186549, MD5: f78ae679294fb0535171ebccb90c780. Below this, there is a note about obtaining the latest blacklist from the site, and a link to the script for download and update.

## 17 URL blacklists

Sites such as `urlblacklist.com` contain ready-made and frequently updated blacklists. The great thing about these lists is that they are categorised. Some scenarios might require a greater sensitivity towards violent material, pornography or pirated software, for example.

## 18 Exception phrase lists

Exception phrase lists are a quick way to unblock material that you do want to give access to. For example, the sites can be unblocked if they include phrases such as 'sexual health'. See the file itself for the format, and carry out some tests using Google to see what works.

```
exceptionphraselist 26
# EXCEPTIONPHRASELIST - INSTRUCTIONS FOR USE
#
# If any of the phrases listed below appear in a web page
# then it will bypass the filtering and be allowed through
# e.g. <medical>
#
# Combinations
# Unblock the page if the following phrases are found on the same page.
# Each line is a new combination.
# e.g.
#<education>,<biology>,<medical>
#
# See the bannedphraselist for more examples.
.
.Include</etc/dansguardian/lists/phraselists/goodphrases/exception>
.Include</etc/dansguardian/lists/phraselists/goodphrases/exception_email>
```

## 19 Add virus checker

If the clients on your network use Windows, it may be good idea to add virus checking of downloaded files. Type `yum install clamd`. Now open `/etc/dansguardian/dansguardian.conf` in an editor and search for the line that begins with 'contentscanner' and

that refers to ClamAV and uncomment it. Start the ClamAV daemon with `service dansguardian start` and then restart DansGuardian.

```
contentscanner = '/etc/dansguardian/contentscanners/clamav.conf'
implementation = 'clamav'
location = '/etc/dansguardian/contentscanners/clamav.conf'
scanner = 'clamav'
scanner_type = 'contentscanner'
scanner_desc = 'ClamAV scanner'
scanner_implemented = 'clamav'
scanner_location = '/etc/dansguardian/contentscanners/clamav.conf'
scanner_scanner = 'clamav'
scanner_scanner_type = 'contentscanner'
scanner_scanner_desc = 'ClamAV scanner'
scanner_scanner_implemented = 'clamav'
scanner_scanner_location = '/etc/dansguardian/contentscanners/clamav.conf'
```

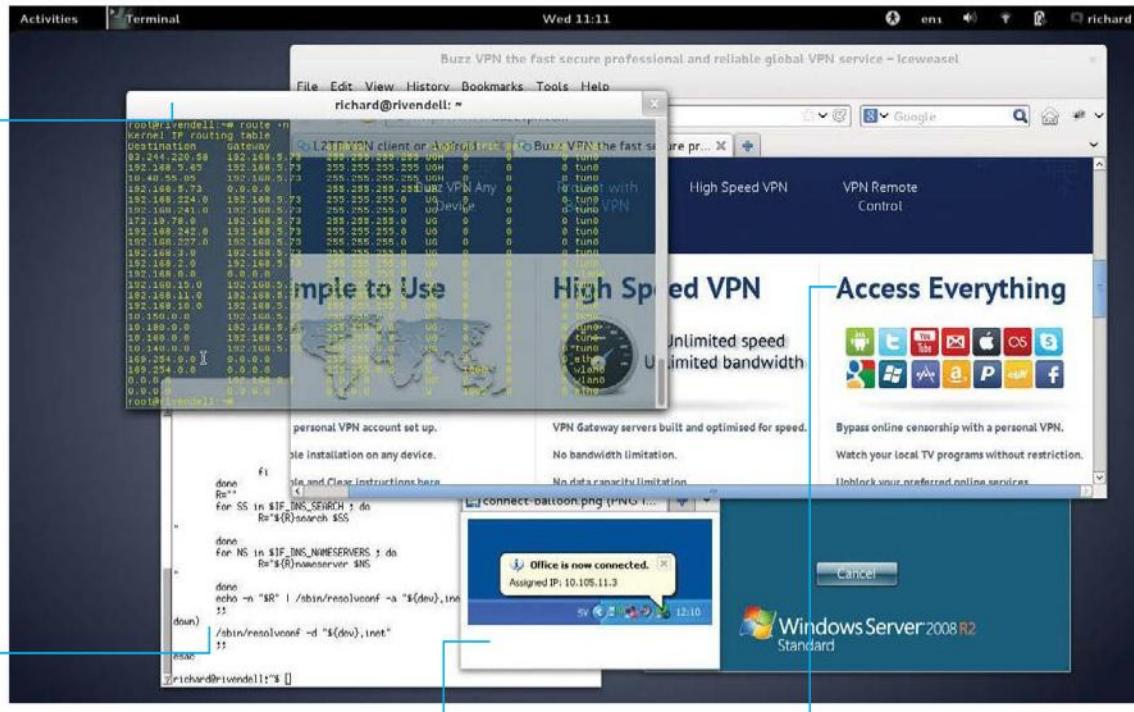
## 20 Add DNS caching

If you are processing requests from a lot of machines, try adding DNS caching to improve performance. You already have a working DNS cache: `Dnsmsaq`, which we installed to provide DHCP. To activate it, edit `/etc/resolv.conf` and make sure that 'nameserver 127.0.0.1' is the first line and that the other nameserver lines refer to a working DNS server. Reboot the machine. Type `dig google.com @localhost` to test that local DNS caching is working.

```
killer@guardian:~$ nano /etc/resolv.conf
GNU nano 2.0.9  File: /etc/resolv.conf  Modified
nameserver 127.0.0.1
nameserver 0.0.0.0
nameserver 192.168.0.1
```

Connect to as many subnets and further VPNs as you would in your office

Push DNS to clients with a resolv.conf updater, as well as IP range and routing information, with simple server directives



Connect and disconnect easily from Windows clients, and admin via GUI

Protect your privacy and keep track of BBC iPlayer from across the globe

# Configure a secure virtual private network

Stop worrying about SSH vulnerabilities and careless users – take control of who connects and how...

**SSH offers astonishing flexibility to create ad hoc tunnels between networks, regardless of any firewall standing in the way.** If this gets you re-evaluating the security of your network, and considering closing off SSH access from outside the network, in favour of restricted access to certain clients only then read on, as we show you how to configure a virtual private network (VPN) to allow only clients with pre-shared credentials to connect to your network.

VPN comes in many flavours, but here we will concentrate on OpenVPN ([openvpn.net](http://openvpn.net)), which tunnels traffic via SSL and combines ease of setup with good functionality and presence across platforms.

While we are on the subject of planned remote connections, you will also want to take a look at VNC, to give users a full remote desktop experience rather than just a remote X Window. This graphical desktop sharing system enables running of software without font issues, for example, and easier access to Windows servers, as well as more complete access to the desktop for certain admin tasks.

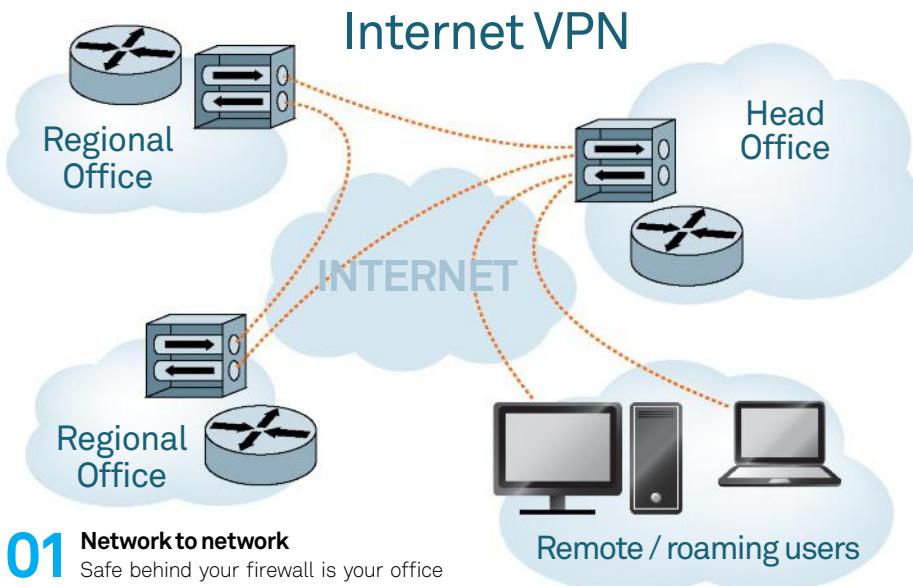
Rounding off, we must mention strongSWAN, which uses the IPSec extensions to encapsulate data securely at the datagram level (OpenVPN uses the good-enough-for-most-purposes OpenSSL – Secure Sockets Layer – library). Essential for the paranoid!

## Resources

**OpenVPN:** [openvpn.net](http://openvpn.net)

**TightVNC** [www.tightvnc.com](http://www.tightvnc.com)

**strongSwan** [www.strongswan.org](http://www.strongswan.org)



## 01 Network to network

**1** Safe behind your firewall is your office network; when you expand to another site, and another network, a VPN allows you to link the two (and further) networks as seamlessly as if they were plugged into the same router, and to give roaming users the same 'local' access.

```
richard@x86: ~
```

File Edit View Search Terminal Tab Help

```
richard@x86: ~
```

```
root@x86:~# apt-cache show openvpn
```

```
Package: openvpn
```

```
Version: 2.3.8
```

```
Architecture: i386
```

```
Maintainer: Alberto González Iniesta <agoinnisti@org>
```

```
Description: an application to securely tunnel IP networks over a
```

```
           single TCP or UDP connection. It can be used to access remote sites, make
```

```
           secure point-to-point connections, enhance wireless security, etc.
```

```
          openvpn uses all of the encryption, authentication, and certification
```

```
          features available in the OpenSSL library (any cipher, key size, or
```

```
          HMAC digest).
```

```
          OpenVPN may use static, pre-shared keys or TLS-based dynamic key exchange. It
```

```
          also supports IPsec with dynamic endpoints (IKEP and IKEv2 clients), tunnels
```

```
          over NAT or connection-oriented stateful firewalls (such as Linux's iptables).
```

```
Homepage: http://www.openvpn.net
```

```
Description:i18n: 200904111849200e16056505c2f
```

```
          A tool for managing and monitoring OpenVPN network servers. It provides a
```

```
          graphical interface, a command-line interface, and a REST API.
```

## 02 OpenVPN

**02** OpenVPN aims to be a universal VPN, and offers great flexibility, but is a relatively small download with few dependencies. It is able to work with passwords, certificates or pre-shared keys, using the OpenSSL library for its encryption capabilities.

```
richard@43: ~
```

File Edit View Search Terminal Tabs Help

```
richard@43: ~
```

```
richard@43: ~
```

```
root@43: ~# apt-get install openvpn resolvconf
```

```
Building package lists... Done
```

```
Building dependency tree
```

```
Reading status information... Done
```

```
The following extra packages will be installed:
```

```
  libpkiutil-helper1
```

```
The following NEW packages will be installed:
```

```
  libpkiutil-helper1
```

```
0 upgraded, 1 newly installed, 0 to remove and 37 not upgraded.
```

```
Need to get 906 kB of archives.
```

```
After this operation, 1,470 kB of additional disk space will be used.
```

```
Do you want to continue [Y/n]?
```

## 03 Easy install

**03** Fire up a terminal emulator and  
apt-get install openvpn as root,

`sudo apt-get install openvpn` if you're on Ubuntu, or `yum-install openvpn` for an RPM-based distro. Add OpenSSL if it's not already on your system, and `resolvconf` may be helpful.

## 04 Address: the problem

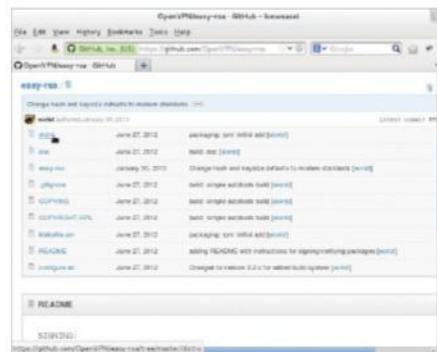
**U4** Before going further, let's consider one potential problem with routing: connecting from an internet cafe using the 192.168.0.0/24 subnet when your network uses the same. Something like 10.66.142.0/24 for your office network could save a lot of grief.

```
richard@x41:~  
File Edit View Search Terminal Tabs Help  
richard@x41:~ richard@x41:~  
root@x41:~# ls /usr/share/doc/openvpn/examples/* --color  
/usr/share/doc/openvpn/examples/easy-rsa:  
    1.0.7  2.0.7  
  
/usr/share/doc/openvpn/examples/sample-config-files:  
client.conf      openvpn-shutdown.sh*  tls-home.conf  
firewall.sh*     openvpn-startup.sh*  xinetd-client.conf  
httpd           openvpn-verb.conf*   xinetd-server-config  
loopback-client  server.conf.gz*    static-home.conf  
loopback-server  static_home.conf*  static-office.conf  
office.up*       static-office.conf  
  
/usr/share/doc/openvpn/examples/sample-keys:  
ca.crt  client.crt  dh1024.pem  pass.key  README  server.key  
ca.key  client.key  pass.crt   pkcs12.p12  server.crt  ta.key  
  
/usr/share/doc/openvpn/examples/sample-scripts:  
auth-pam.pl*    bridge-stop*    ucn.pl*  
bridge-start*   openvpn.init.gz* verify.cn*  
root@x41:~#
```

## 05 Simpler config

**Q5** How do you keep a flexible app simple to configure? By including config examples to

modify. Grab the easy-rsa examples with  
sudo cp -R /usr/share/doc/openvpn/  
examples/easy-rsa /etc/openvpn/



## 06 Public-key infrastructure (PKI)

**88** We're going to use easyrsa to create a master CA certificate, to sign the certificates which we'll generate for the server and each client. Recently easyrsa has been separated out from OpenVPN, so you may need to download it from [github.com/OpenVPN/easy-rsa](https://github.com/OpenVPN/easy-rsa)

```
mc [root@x41]:/etc/openvpn/easy-rsa/2.0
e Edit View Search Terminal Tabs Help
richard@x41: ~          mc [root@x41]:/etc/openvpn/easy-rsa/2.0/var
GNU nano 2.2.6      File: /etc/openvpn/easy-rsa/2.0/var
port CA_EXPIRE=3650

In how many days should certificates expire?
port KEY_EXPIRE=3650

These are the default values for fields
which will be placed in the certificate.
Don't leave any of these fields blank.
port KEY_COUNTRY="GB"
port KEY_PROVINCE="Merseyside"
port KEY_CITY="Liverpool"
port KEY_ORG="World Domination"
port KEY_EMAIL="me@"
port KEY_EMAIL=mail@host.domain
port KEY_CN=changeme
port KEY_NAME=changeme
port KEY_OU=changeme
port PKCS11_MODULE_PATH=changeme
port PKCS11_PIN=1234

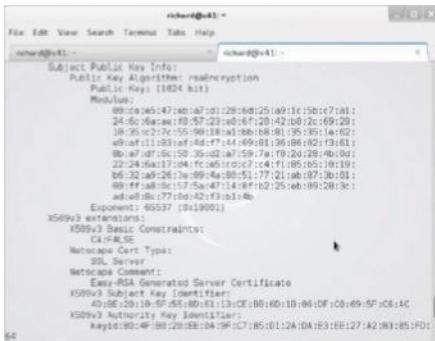
Get Help F2 WriteOut F3 Read File F4 Prev Page F5 Cut
D1 F6 Copy F7 Paste F8 F9 Next Page F10 F11
```

## 07 Master certificate

**07** Edit the vars files, changing the KEY\_COUNTRY, KEY\_PROVINCE, KEY\_CITY, KEY\_ORG and KEY\_EMAIL parameters. Other values that may need changing are usually helpfully marked as "=changeme" – both the comments and the README file will guide you.

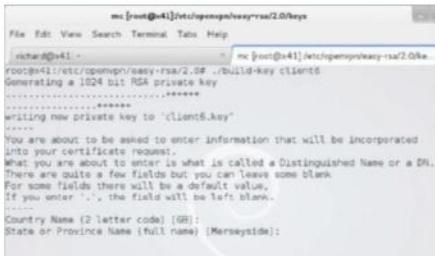
## 08 Generation game

**08** From within the same directory as the vars file we have just generated – /etc/openvpn/easyrsa/2.0/ in this case – we run the build script. Note that instead of ‘hostname’ for Common Name, you may wish to enter OpenVPN-CA.



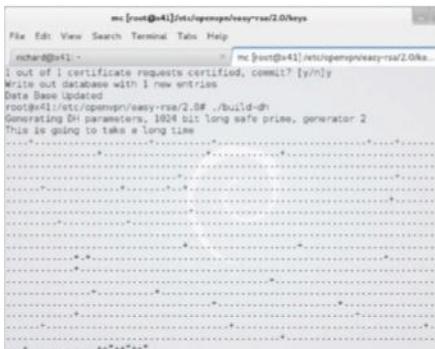
## 09 Build server certificate

**09** Running `./build-key-server server`  
next differs slightly as 'server' is offered as  
the Common Name (accept this), then you are  
offered a challenge response (skip this), and to  
sign the certificate (choose yes).



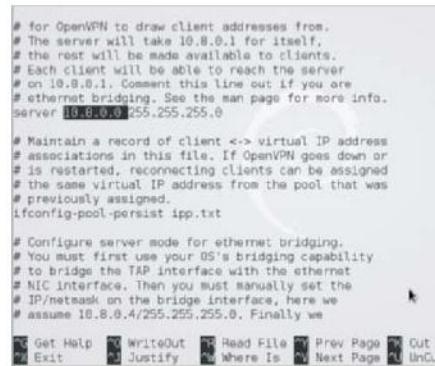
## 10 Roll out the client certs

**10** Now build as many client certificates as you need with variations on `./build-key client1` – because each client certificate is signed with the same master certificate as the server key, the server will not need to keep copies of the client keys.



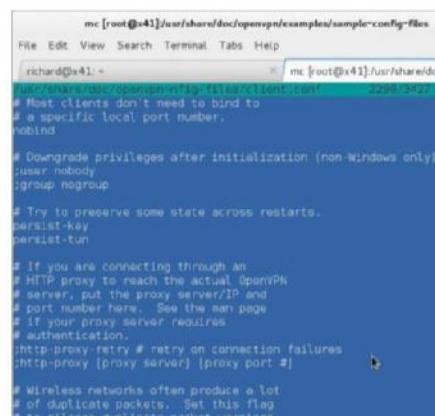
## 11 Diffie–Hellman

No, it's not a brand of mayonnaise! The Diffie-Hellman key exchange method "allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel." Run `/bu11d-dh`.



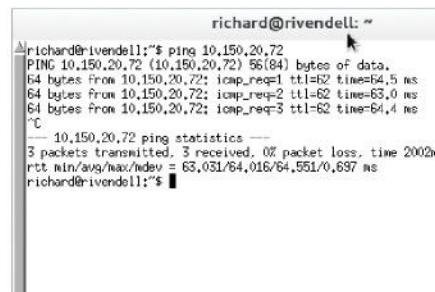
## 12 Server config

**T2** Start with the sample **server.conf** from the `/usr/share/doc/openvpn/` example configs. Change the address range from 10.8.0.0 to your own. Other options include the ability to push the route, eg: **push "route 10.13.101.1 255.0.0.0"**



## 13 Nearly there

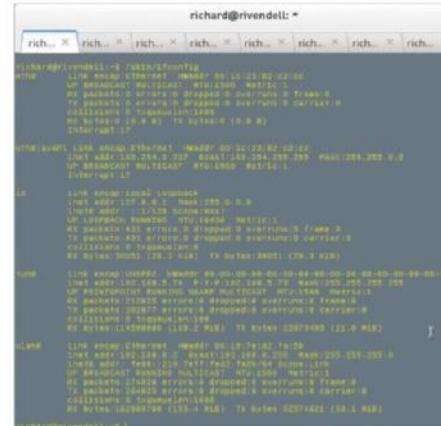
**13** On your client PCs, copy the keys you have generated (using scp or a USB key), and edit the sample **client.conf** file. Uncommenting the **user nobody** and **group nobody** directives will add to security. Now it's time to test...



## **14** Is anyone there?

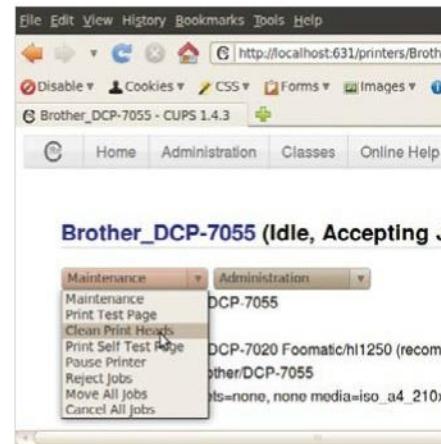
**14** Start OpenVPN on the client with `openvpn path/to/conf`. From the client, try

pinging an address on the remote network. Given correct address data, any errors are likely to be firewall-related. Success? Now start with `/etc/init.d/openvpn start`



## 15 We have tunnel!

**13** ifconfig tun0 (or ifconfig tap0 if you're using a virtual Ethernet device instead of a point-to-point IP tunnel) will now show all the info, giving the addresses at each end of the tunnel. If you enabled the **push "route..."** and **push "route-gateway..."** directives in the server config, you will now be able to also reach whatever other networks are visible to the server via other VPNs, as shown in the opening screenshot of the article. The **push "dhcp-option DNS 10.66...."** directive may also be useful to you.



16 Remote access

**10** Now you have your secure connection into the office, you'll want to do more than just ping boxes. You can roam the intranet, performing local admin tasks on printers and servers from the comfort of your favourite cafe...



## 17 Desktop sharing

Adding VNC into the mix will enable you to work with GUI apps on remote systems across the VPN, whether GNU/Linux, Windows or whatever. xvnc4viewer will give you more power than Ubuntu's built-in rdesktop, and TightVNC at both ends gets through narrow bandwidth connections.



## 18 Spread the network

As well as clients for UNIX, Windows and even Maemo, there's an Android port of the client software at [github.com/fries](https://github.com/fries). Once upon a time OpenVPN was console-only admin on Windows, unless you went to [openvpn.se](http://openvpn.se); now it's all included in the package.

```
#!/bin/sh
# File: /etc/openvpn/server.conf

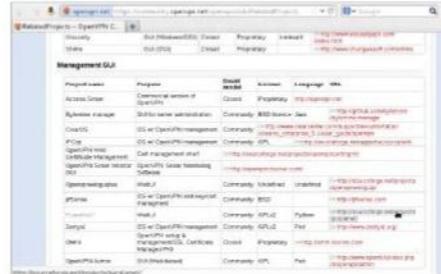
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface.
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
dev tun

management localhost 4202

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel. If you
# have more than one, on XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
```

## 19 Admin tasks

Enabling management on the port of your choosing gives you access via **telnet localhost:4202** – from here you can disconnect clients; toggle logging; and perform tests and debugging. The management GUI accesses OpenVPN through this interface.



## 20 GUI choice

OpenVPN's popularity can be seen in the vast choice of third-party GUIs, both to OpenVPN itself (connection clients) and to the management interface. While proprietary bolt-ons are a familiar tale, FOSS options are available too.



## 21 Hassle-free VPN

If you just wanted a VPN to protect your browsing privacy, say, or to catch BBC iPlayer while overseas, then one of the many commercial VPN providers is a hassle-free alternative, with downloadable clients for nearly every device. Read the reviews to find a suitable one.



## 22 Security first

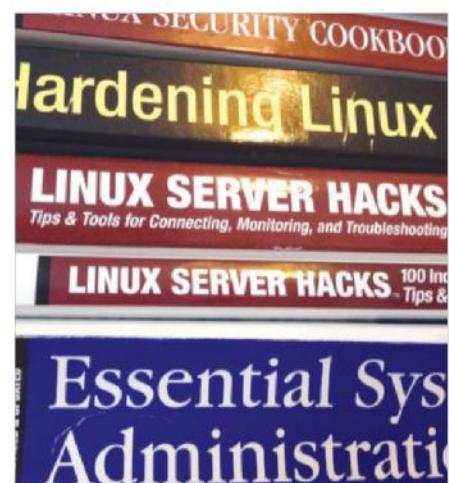
Alternatively, IPsec gives you secure encapsulation of your data inside an IPsec packet, aiming for authentication, integrity and confidentiality. It's favoured by government agencies, those fearing industrial espionage, and anyone else feeling justifiably paranoid.

**“Remote access with VPN saves opening networks to SSH tunnel’s firewall-defying antics”**



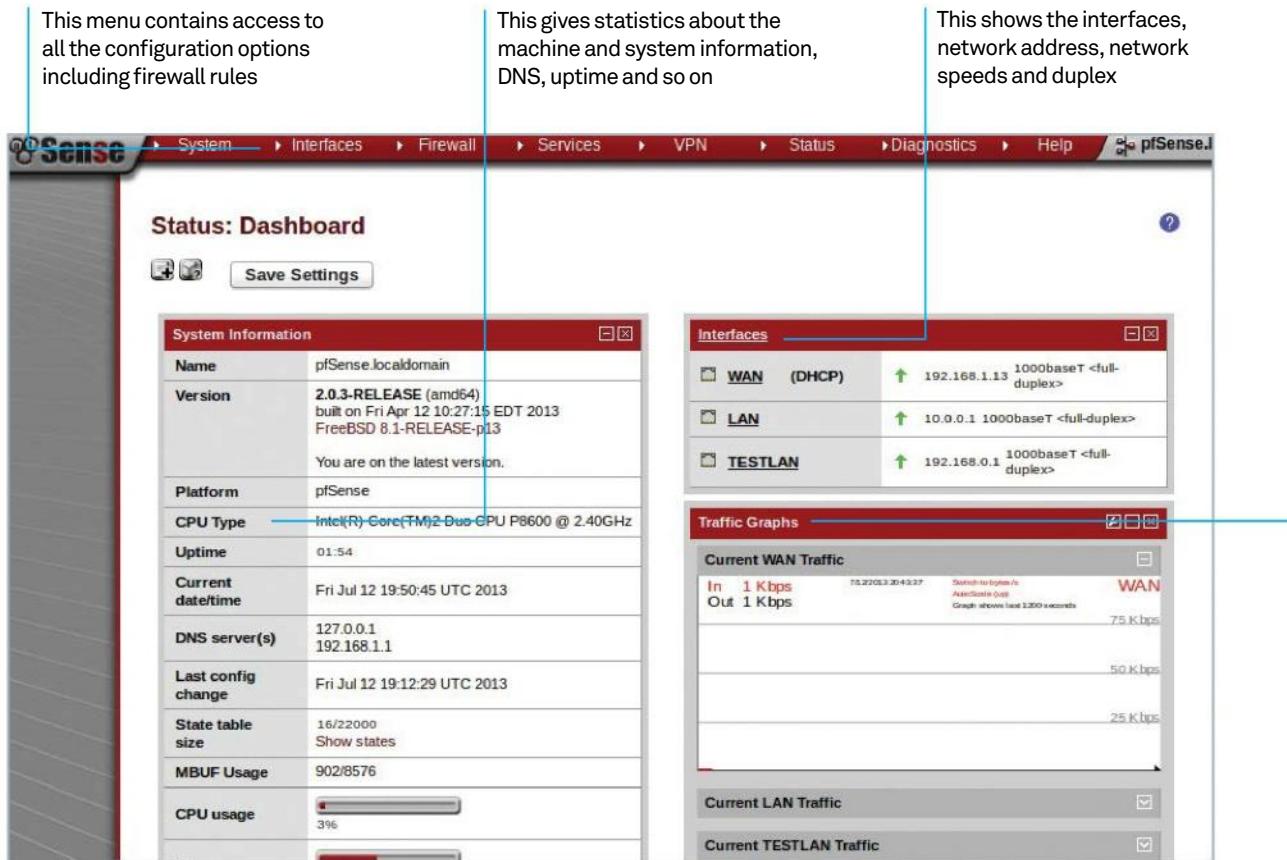
## 23 Swanning through

For IPsec, strongSwan – a successor to FreeS/WAN (Free Secure Wide-Area Networking) – provides compatibility with other IPsec implementations, including clients on other platforms, combined with IKEv1 and IKEv2, and a good reputation for security.



## 24 Brain food

There's plenty of accumulated wisdom on remote access and admin. While the world wide web offers much that is useful, don't neglect print format! Some of the sysadmin manuals and server hacks books available contain some great tips for remote, secure admin and much more.



# Build your own pro-grade firewall

Learn how to create a powerful multi-network hardware firewall with a redundant computer

## Resources

A Linux PC with 3 network cards (min 300MHz, 128MB RAM)

pfSense live CD: [www.pfsense.org](http://www.pfsense.org)

Labelling system for network cards

An ADSL or cable modem

A second Linux PC

This in-depth tutorial covers setting up a hardware-based firewall and configuring it to make it hacker resistant and business class.

It will cover the configuration of a basic two-network setup consisting of an internal network for all your test setups and a second LAN that can be used for normal everyday usage. We will include a DHCP setup on your second LAN to make your life that little bit easier.

The networks are to be configured in such a way that any breakages on your test network won't affect your normal network. This guide will

also cover creating a sensible rule base to which you can add extra rules if you wish. Additionally, you'll find tips and tricks to make everything more secure than a simple default setup. Finally, we will cover how to back up and restore your firewall configuration, should the worst happen.

If you want to just experiment with this without going the whole hog, you can do it within a virtual machine, two virtual networks and a bridged adaptor to your local network. The scope of this setup is outside the bounds of this article, but our walkthrough should still work perfectly.

The basic network I/O occurring through your firewall

```
Welcome to pfSense 2.0.3-RELEASE ...

Mounting unionfs directories...done.
Creating symlinks.....done.
Launching the init system... done.
Initializing..... done.
Starting device manager (devd)...done.

[ Press R to enter recovery mode or ]
[ press I to launch the installer ]

(R)ecovery mode can assist by rescuing config.xml
from a broken hard disk installation, etc.

(I)nstaller may be invoked now if you do
not wish to boot into the liveCD environment at this time.

(C) continues the LiveCD bootup without further pause.

Timeout before auto boot continues (seconds): 4
```

## 01 Install pfSense on your redundant PC

Boot from the pfSense live CD you downloaded and burnt in the prerequisites (see Resources). Allow it to boot up with defaults until you get to the screen that mentions recovery and installer. Press the I key to invoke the installer. Accept the defaults presented on screen by selecting 'Accept these defaults'. The only possible change you might want to make is to your keyboard layout if you have a non-US/UK-type keyboard. Now simply select Quick/Easy Install. Read the warning – the installation will totally destroy any information on the disk, so back up first if you want to preserve your data. When you're ready, select OK. Once the installation is done, select Standard Kernel and once that's configured, navigate to Reboot and press Enter. Make a note of the default username and password (admin/pfsense). Remove the CD and the machine should reboot into the network configuration menu where all the good stuff starts to happen.

## 02 Configure networking

At this point, make sure your network cables are not plugged in. After booting into pfSense you will see a basic text configuration screen and a list of the network cards installed. When asked if you wish to configure the VLANs, select no (by pressing N). Next we are going to auto-detect the network. To set up the WAN connection, press A. Now insert the WAN cable from your router into the first network port. You will see it change status to UP, then press Enter to continue. We have now configured the WAN port to the internet – repeat the same process for your first and second LAN cards in the same fashion. Once complete, press Enter to continue.

This finishes the installation and lets the firewall know there are no more network connections to be configured. Answer Yes

when asked 'Do you wish to proceed?'. It will now commit the settings to disk. It will also give you a list of networks to match up again your network cables. It is a good idea to label them up now to save confusion later.

## 03 Introducing the pfSense setup

After configuring the network connections and rebooting, you'll still see the CLI with a series of menu options. Since the other networks need to be configured and you can do this by pressing 2 on the console. You'll now see you can configure IP address setup for all the networks. Select the NIC that corresponds to your wireless or basic internal network. This is our (WIRELESS) LAN so let's give it 192.168.1.1 with 254 addresses. Enter the IP 192.168.1.1 –



```
Welcome to pfSense 2.0.3-RELEASE ...

Mounting unionfs directories...done.
Creating symlinks.....done.
Launching the init system... done.
Initializing..... done.
Starting device manager (devd)...done.

[ Press R to enter recovery mode or ]
[ press I to launch the installer ]

(R)ecovery mode can assist by rescuing config.xml
from a broken hard disk installation, etc.

(I)nstaller may be invoked now if you do
not wish to boot into the liveCD environment at this time.

(C) continues the LiveCD bootup without further pause.

Timeout before auto boot continues (seconds): 4
```

this will become our gateway. This tutorial is using a /24 network, so type in 24 followed by Enter. It will ask if this network needs a DHCP server – select Yes. The configuration program will then ask about the start of the DHCP range. It's best to start at 192.168.1.2. Follow this with the end of the range, 192.168.1.32. This is up to you and depends on your needs, but 30 DHCP leases is more than enough. Press N on the HTTP protocol question. Repeat the process with the other network and select 10.0.0.1 as the interface address, 24 as the network mask and use the range 10.0.0.2 – 10.0.0.32.

## 04 Using the pfSense GUI

In this section we'll set up the basic GUI. Connect a laptop to the network of the WIRELESS LAN and open a web browser and enter <https://192.168.1.1> in your browser. You may receive a warning about an untrusted network connection, but that is fine to ignore for our purposes. This address and webpage is the network address (gateway) you configured earlier in the tutorial. It may be necessary to add an exception and hit Continue on your web GUI page.

You will be greeted with the setup wizard. Select Next to get started. At this point you can leave the hostname and network name alone, unless you want to put your own DNS servers in. If you leave the override DNS feature, you will get your DNS for your DHCP servers from your ISP.

Configure the time servers and click Next. On the next page you can configure any extra setup information if your ISP requires it. Click Next to go to the LAN page. Lastly, change the admin password to a secure one of your choice. At this point the firewall will reload its rules. Enable the third network, click Interfaces>OPT1 and select 'enable interface' and click Save. Rename OPT1 to LAN by clicking on Interfaces>OPT1 and renaming it LAN.



## 05 How to create a basic rule

All rules are added in the same way; just add and modify each rule to fit the requirements. Click the bottom left '+' symbol from the Firewall Rules page to start creating one. Now we can add web browsing. Set action to pass (unless you wish to set up a rule to drop traffic). Choose your source interface (LAN/WIRELESS). Follow this by selecting your protocol to use (usually TCP, but things like DNS require UDP port 53). On the next item, select the destination. Usually this will be the any address for external traffic and WIRELESS or LAN subnet or address, depending on requirements.

Destination port is straightforward enough: you can select a range of ports by either using the drop-down menus or entering your own ranges (for now, just select HTTP). Using multiple ports is covered later in the article.

One set of rules definitely needed for both networks is basic HTTP and HTTPS rules for browsing. You will also want to implement a 'drop all' rule. As the name implies, this drops all traffic. This makes sure no traffic escapes out of your network that you intended. To do this, just set up a rule that has drop for the action, networks and port ranges set to any TCP/UDP on the protocol. Do this for both networks.

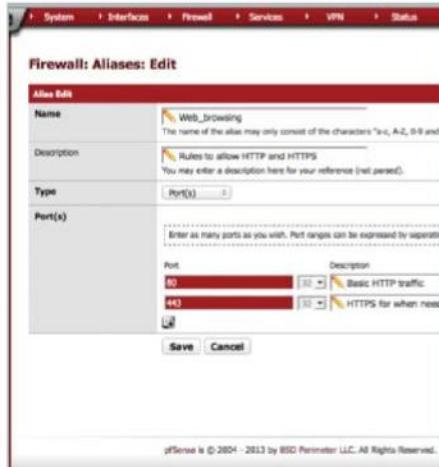
## 06 Aliases make life easier

Aliases enable you to group ports together. As the name suggests, they allow you to use an alias in your rules that can refer to groups of items. An example would be combining HTTP and HTTPS together in one alias. No need for multiple rules – just one alias can be used to ensure correct ports are opened!

From the Firewall menu, select Aliases. Use the '+' on the right. To implement HTTP and HTTPS together, give it a name like Web\_browsing\_ports – ensure it is descriptive. Select ports from the Type drop-down. Hit the

'+' button below the ports and add 80 in the port and HTTP in description. To add HTTPS, click the '+' button, but use port 443. Save and apply changes. Aliases are not limited to ports, but can also be used for hosts and networks. To implement an alias in a rule (assuming the alias has been created beforehand) go to the Rules Port drop-down, select Other and begin to type the name of the alias. It should pop up a list. Click on the alias needed and accept. Apply the changes once the rule is created. Similar rules can be created between networks. An example is SSH. Implement this rule the same way.

# "No need for multiple rules – just one alias can be used to ensure correct ports are opened"



## 07 Enhanced rule sets

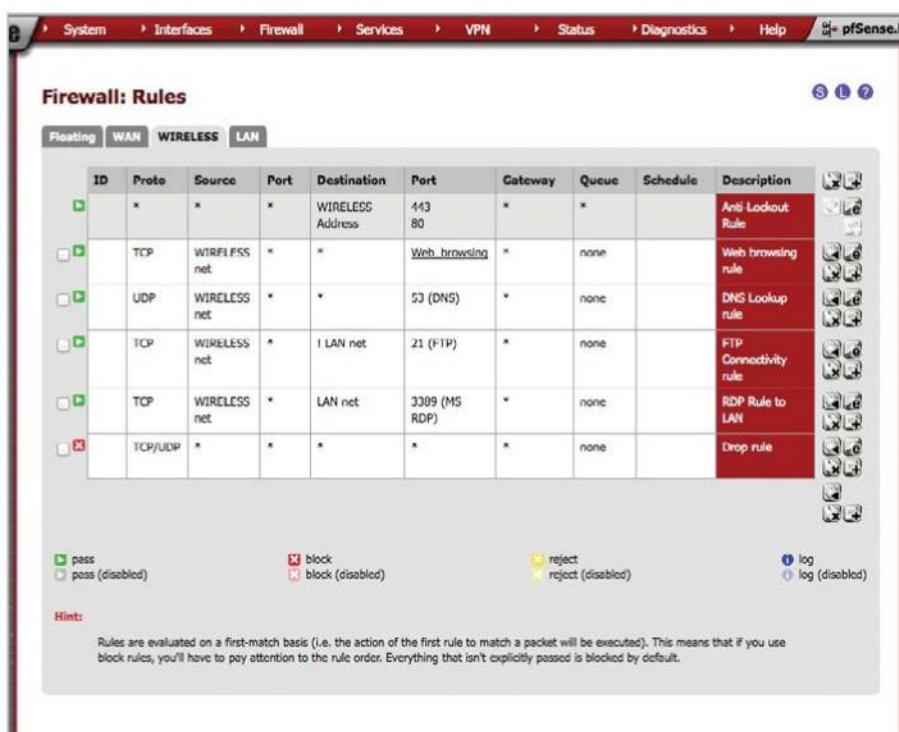
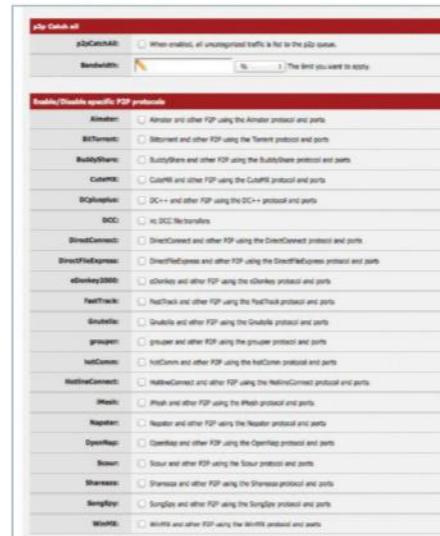
Now that you understand how basic rules work, it is time to group together a more enhanced rule set. As a minimum, set up both networks to have the following flowing out the internet. HTTP and HTTPS (remember to use an alias here!), include FTP, DNS (using UDP) as well as SSH if needed. However, box clever here. If you only use SSH to talk to a specific number of hosts, use an alias with the Hosts drop-down and enter the IP addresses into the alias. That way, should a machine be compromised, it will

not be able to talk SSH on port 22 to anything but those boxes defined in the alias. The more specific the rules, the more secure they are. You will also need to repeat the process on the LAN, assuming you want the same rights. To prevent a network talking to another on a certain port and protocol, use the NOT option in the rule base. An example would be to change the web browser rule to say destination NOT LAN – you will then find you can no longer browse any web server on the test network, but can browse the internet.

## 08 Managing the bandwidth

Now we can look at some other features such as bandwidth management. PfSense makes it easy to block file-sharing platforms such as BitTorrent, WinMX and similar. It can also split the bandwidth between the two networks. Do this by going to Firewall>Traffic Shaper. Click the Wizards tab. There are a number of different scenarios; select the 'Single WAN, Multi LAN' option. Enter number of LANs (two in this case) and press Next. Fill in your available download and upload speeds. Leave the other components and click Next. Unless you use SIP, click Next. Penalty box can be used to restrict specific groups or alias groups of machines to a percentage of the capacity if needed. Click Next. Use this page to lower the priority or even block P2P traffic completely.

Click Enable on the Traffic Shaper wizard and then select any protocols to allow/block. Edit to the preferred setup and then click Next. On this page, configure traffic shaping for games, with preconfigured optimal setups if needed. Finally you can do the same for applications if you wish to, such as RDP, VNC etc. Click Finish. To remove the shaping, go back to the Firewall Traffic Shaper menu and select 'Remove shaper'.



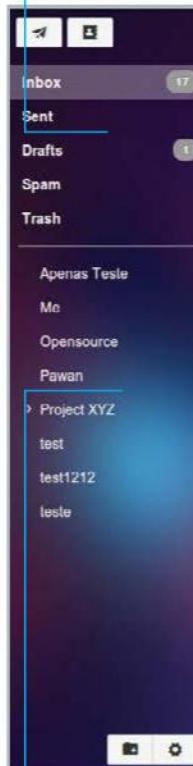
## 09 Turn on logging

Sometimes, rules don't actually do what you planned, but there are a number of tools for logging and manipulating rules. It's wise to be able to review the logs to see exactly what's going on. To turn logs on, simply go back into the Rules menu, find the rule that you think may be problematic, and tick the 'Log this rule' box. Don't forget that rules are evaluated on a first-match basis; so, for example, having the drop all rule before the rule trying to be tested would mean the rule would never get evaluated.

Backing up is also an important exercise and very simple to execute. Go to the menu, select Diagnostics>Backup/Restore. The options on this page are simple enough. It is recommended to tick the box to encrypt the backups. Give it a good password that you will remember. We also suggest you leave the box 'Do not backup RRD data' selected. This is just performance data and isn't really needed day-to-day.

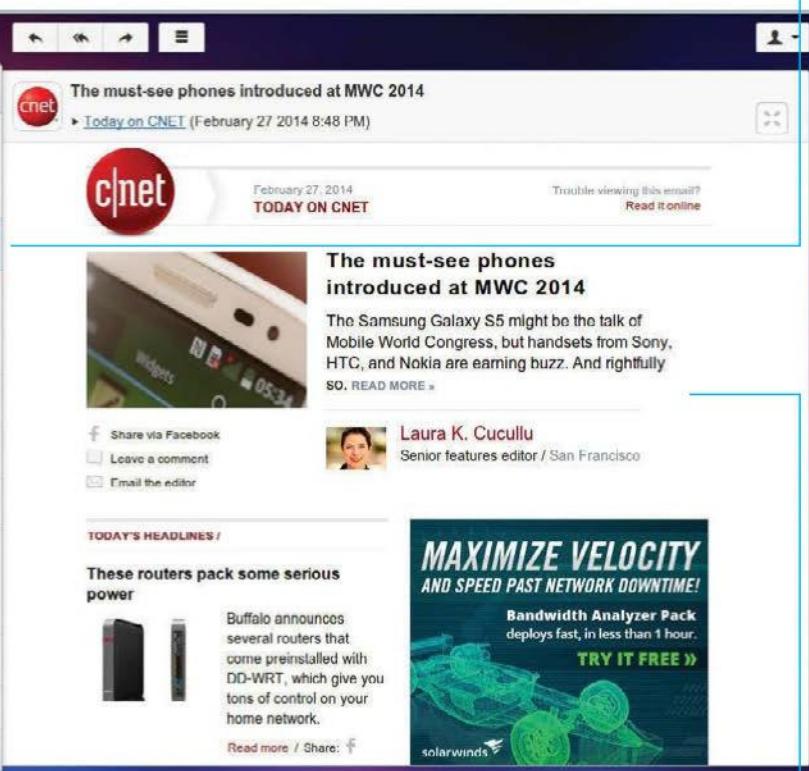
Should the firewall ever need rebuilding from scratch, you will have to redo the steps right up until you have the GUI. The Restore menu, found in the Diagnostics menu, has the checkbox to restore from backup, but also the option to only restore parts, such as the rule base.

A traditional mail client layout exists in Rainloop that connects with the folders and emails of your server



Customise your experience with different folders, extra accounts and even social network login support

Open and favourite emails are remembered between sessions so your unread accounts are accurate wherever you log in



The preview can be made fullscreen in the same window, instead of opening a different page or tab – this reduces server load

# Host your own webmail server

Cut out the middleman by managing your own webmail for personal accounts and avoid any unnecessary downtime

**While using webmail may be incredibly convenient, you're also at the mercy of another company's server and privacy policies.** With the way that people are connected online today it's almost impossible to go back to the mail client system of old, even if security and privacy are far superior to Gmail.

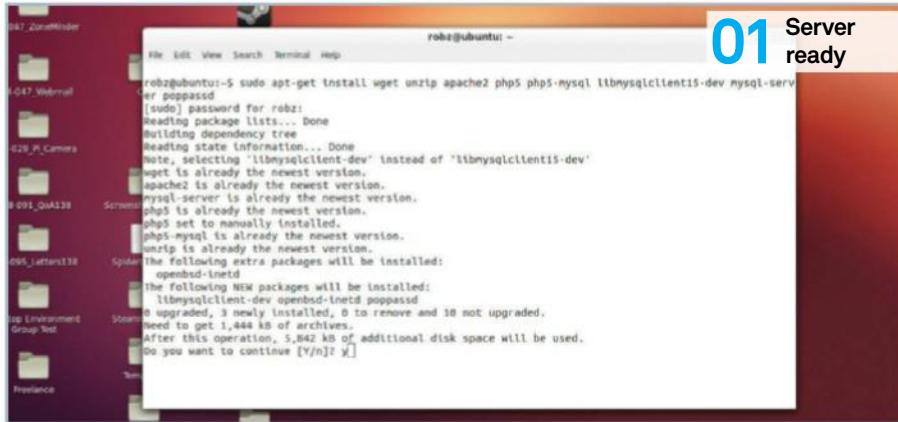
But there is another solution that satisfies both requirements; by hosting your very own webmail server you can have both the convenience of worldwide access while having

the privacy of a desktop mail client. By using Rainloop, you can quickly and easily set up your own webmail server with your own custom settings and email addresses.

You will need a server or always-on PC in order to host your webmail, otherwise it will only work when your computer is actually on. Be aware that it may also increase your bandwidth usage on a monthly basis, so don't send huge files over it unless you need to. Interested? Let's get going.

## Resources

Rainloop <http://rainloop.net/downloads>  
A server



## 01 Server ready

Before we even begin to look at Rainloop in depth, we need to make sure the system hosting it is server ready. Install the following packages from the terminal:

```
$ sudo apt-get install apache2 php5
php5-mysql libmysqlclient15-dev mysql-
server popasswd
```

Some or most of these packages may already be installed, but it's worth checking.

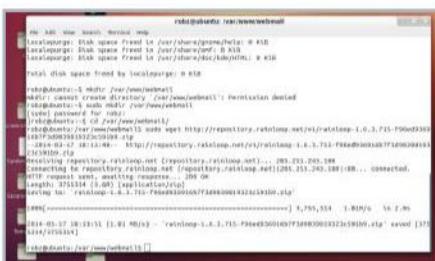


## 02 Set up directories

We'll need specific directories created to get Rainloop to work. You can do this in a file manager, but it works in the command line like so:

```
$ sudo mkdir /var/www/webmail
```

Move to the directory for the next step, as this is where all our Rainloop files will live.



## 03 Download Rainloop

Now it's time to get Rainloop. If you have a graphical interface, download the latest version of Rainloop from the website and unzip it.

## 01 Server ready

Otherwise, use wget to download it:

```
$ wget http://repository.rainloop.net/v1/
rainloop-[current version].zip
```

And then unzip it to proceed.

## 04 Elevate permissions

Open the terminal if it's not already and use cd to move to /var/www/webmail. Run the following two commands to elevate the permissions of the necessary files:

```
sudo find . -type d -exec chmod 755 {} \;
sudo find . -type f -exec chmod 644 {} \;
```



## 05 Final permissions

Finish off making sure all the permissions are set by running:

```
sudo chown -R www-data:www-data /var/
www/webmail
```

Rainloop is now just about ready to go, and most of the rest of the setup will occur within the interface itself.

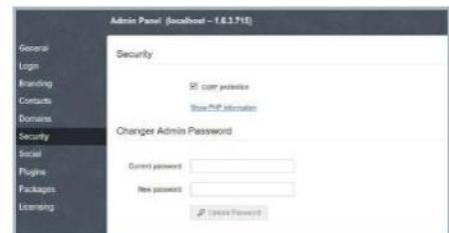


## 06 Access Rainloop

Now what we have to do is log into Rainloop from a browser; you do this either through the IP of the server or its web domain if you know it. If you're accessing it on a local

machine, use 127.0.0.1. Enter the following into your address bar:

[http://\[IP address or domain\]/webmail/?admin](http://[IP address or domain]/webmail/?admin)



## 07 Log in

The username and password for this default version of Rainloop is 'admin' and 12345 respectively. Log in to access the interface; our first task is to then change the default password. Create a secure password using standard password etiquette, as the URL is quite common.



## 08 Add domains

In the Admin Panel, find domains and then add a domain. Here you can add your own personal email or work domains and use the IP or server address in either or both of the IMAP and SMTP fields. You don't need to add Gmail or a handful of other web services, as they're already listed.



## 09 Domain ports

Make sure the ports are correct on the server addresses: for a local server, the default ports should be fine. Also, make sure that 'Use short login form' is checked, name the server as the @ address of your email (eg example.com) and then click Add to save it to the list.

## 10 Contacts database

To support contacts, we need to add a MySQL database. Open up the terminal again and type in:

```
$ sudo mysql -u root -p
```

Enter your password and you'll be dropped into the MySQL shell to create the database.



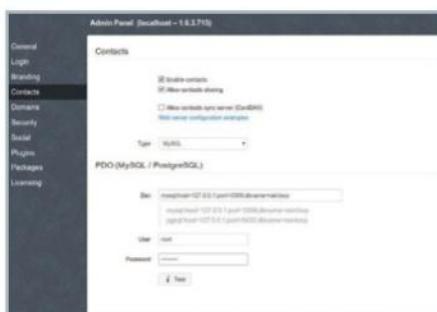
## 11 Create the database

To create our contacts database, enter the following command:

```
create database rainloop;
```

This, as you might have guessed, simply creates the database named rainloop. To confirm the operation and quit out of the MySQL shell use:

```
exit;
```



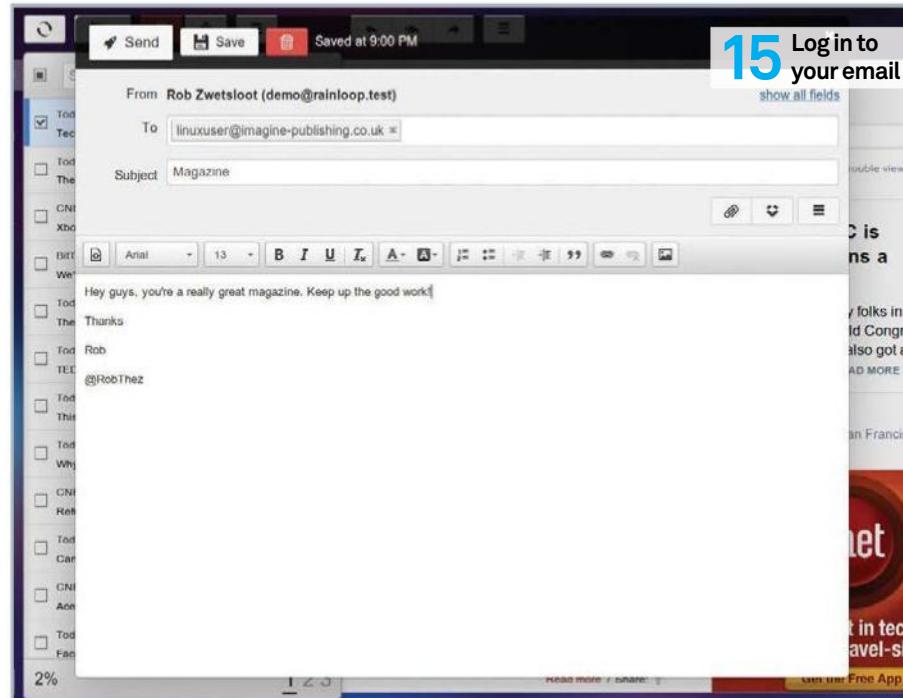
## 12 Enable Contacts

Back in the admin panel on Rainloop, go to the Contacts tab and check the Enable Contacts box. Below that, change the type of the database to MySQL. You'll then also need to add the username and password for the MySQL server.

## 13 Password changing plug-in

You'll need to add a specific password change plug-in to be able to change any passwords on a Linux mail-server. To do this, open up the terminal and first install the plug-in to your system with:

```
$ sudo apt-get install poppasse
```



**“It’s almost impossible to go back to the mail client system of old, even if security and privacy are far superior”**



## 14 Activate password plug-in

Now you need to add the plug-in on the admin panel of Rainloop. Go to the packages tab and find the plug-in on the list; activate it by clicking the arrow next to release date.

## 15 Log in to your email

To log in to your webmail, go to the webmail address for your server:

[http://\[IP or domain\]/webmail](http://[IP or domain]/webmail)

And use your normal login details for the mail

## 15 Log in to your email

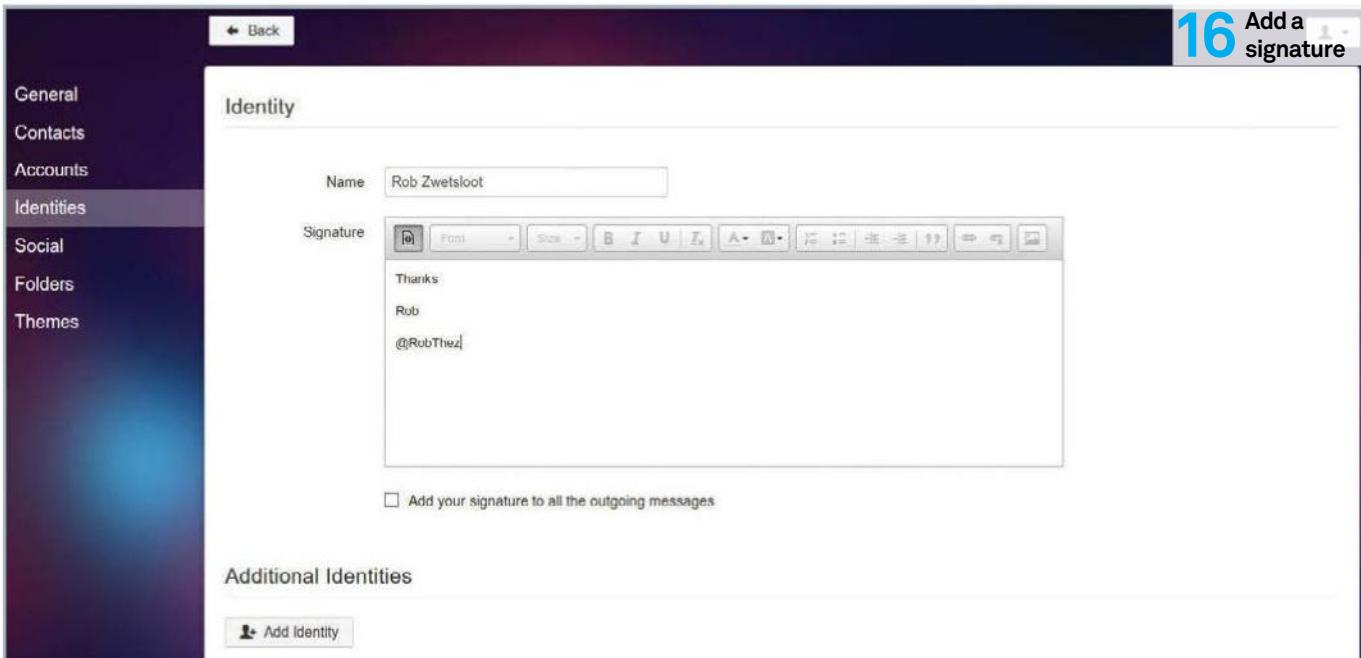
address you've added to the server. It will log you in and work just like any email client, with a column of emails on the left and a preview pane located on the right. You can then expand the emails to take up fullscreen.

Rainloop will remember which emails you have read and connect to drafts and sent folders whenever you log in.



## 16 Add a signature

Click on the gear symbol at the bottom to access the Settings. From here you can create an identity, each with an individual signature. This is useful for if you're using multiple accounts, or if multiple people are using the same email.



## 16 Add a signature

Located next to the settings cog on the main screen is the new folder button. You can add folders and then move emails to them for better filing, and these will sync with the server. You can also have parents and children of folders in your hierarchy



## 18 Manager folders

In settings you can create folders, but you can also delete, hide and edit the folders too. There's also an option to change the system folders, allowing you to filter spam, sent or other types of messages to pre-determined folders. This is currently the only way to filter email.

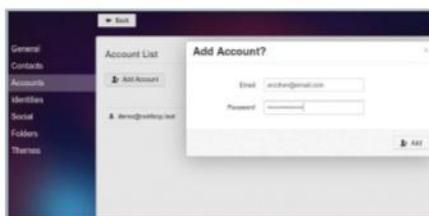
## 19 Connect via social media

Rainloop has an interesting feature of allowing you to log in to your webmail for specific accounts via Twitter, Facebook or Google sign in. Before you have a go at this, make sure that you're signed in with the webmail account you want to activate this on.



## 20 Choose your account

Once you're logged in, go to the Settings menu and locate the Social tab. In here you can choose between the three social media types – you basically need to log in under one or all of them here first before adding the feature to the main interface.



## 21 Add a new account

You can add another account to your login for that email address, as long as it's included in the accounts you have created on the admin page. This way you can consolidate all of your

emails into one single login. This can be done from the main page, or more preferably from the Accounts tab in the Settings menu.

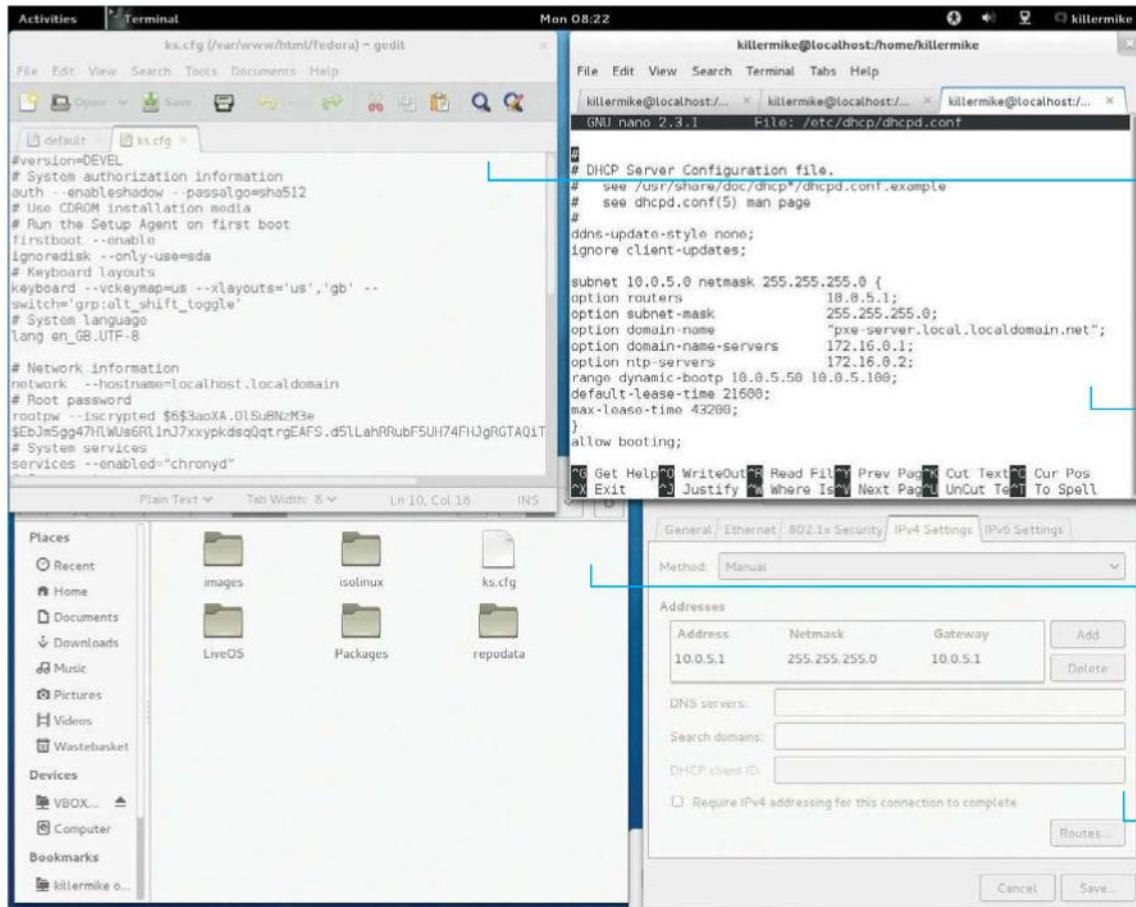
## 22 More on social

Having your social media work as a log on means that you can only use them on the one account that you've set them up for. It also means that people can more easily log in to your accounts if they have access to your regular system and you don't log out.

Admin Panel (localhost - 18.3.719)		
• Slack API	1.0	11.12.2013
• WhatsApp API	1.0	11.12.2013
• Google Analytics	1.0	28.10.2013
• reCAPTCHA	1.0	12.11.2013
• Change password (OpenID)	1.0	28.10.2013
• Change password (Facebook)	1.0	23.11.2013
• Change password (Twitter)	1.0	14.03.2014
• OpenID SSO Credentials	1.0	28.01.2014
• Standard sign-on service	1.0	28.11.2013

## 23 More to come

There are some features missing at the moment that you might miss from some regular clients – Out Of Office messages and automatic filters, for example. These features will eventually come, whether officially or via plug-ins, so keep an eye out for updates.



The Kickstart file provides all of the answers for the installer so that it doesn't have to prompt the user

A boot server needs to run a properly configured DHCP server so that connected machines boot from it

We'll place the contents of an ISO DVD into a folder that is served over HTTP

In this example, we shall set up a boot server that has two network cards – one for internet access and one for installation targets

# Deploy Fedora over a network

Learn how to install Fedora to an entire LAN



## Resources

**Working Fedora box**

**Two network adaptors**

**Fedora installation DVD ISO**

**Network of at least two machines**

**Installing Linux on a single box is easy, but try extending that to a room, or even building, full of computers and you'll face a massive headache.** To save you from running back and forth between all those computers, we'll show you how to set up an automated network install.

This project has two main stages. Firstly, a working boot server must be established. Secondly, a Kickstart file must be created to satisfy the installer and ensure that it does not require any interaction from the administrator.

Some solutions of this type favour making a bootable respin of the installation medium, but

the problem with that method is that it becomes extremely tricky to make changes to the setup on installation day. So instead we're going to look at an approach that works from within a normal Linux installation, Fedora in this case. If you need it to be portable, no problem – just install Linux to a flash drive and work from that.

It ought to go without saying, but be a bit careful when connecting the server up to the switch/router of the clients. When fully configured, this machine will happily wipe and configure anything with which it comes into contact.



## 01 Install Linux

These examples use Fedora Linux, but any Red Hat-derived distro should work. If you install to a removable medium, make sure that you have enough free space to make a copy of the installation DVD ISO. 16GB of free space is a sensible minimum.

```
File Edit View Search Devices Help
--> Package dhcpcd-12.4.2.5-7.fc18 will be installed
--> Package tftp-1.0.66-9.5.2-6.fc18 will be installed
--> Package tftpd-server-1.0.66-9.5.2-6.fc18 will be installed
--> Package xinetd-1.0.66-2.12.3.35-4.fc18 will be installed
--> Finished Dependency Resolution
Dependencies Resolved

Package          Arch      Version       Repository      Size
dhcpcd           RPM      12.4.2.5-7.fc18    updates      510 k
tftp              RPM      5.2-6.fc18      Fedora       34 k
tftpd-server     RPM      5.2-6.fc18      Fedora       43 k
xinetd           RPM      2.12.3.35-4.fc18   Fedora      129 k

Transaction Summary
Install 4 Packages
Total download size: 715 k
Installed size: 1.0 M
Is this ok [y/N]:
```

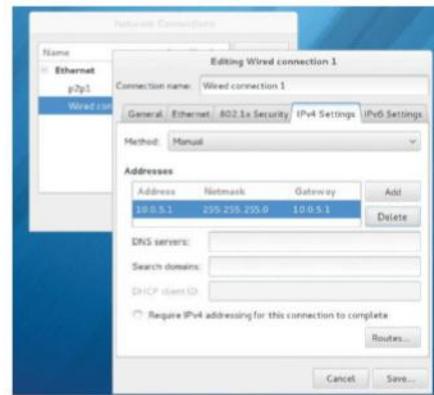
## 02 Add packages

Make yourself root (type `su` into a terminal), then use YUM to add to extra packages with `yum install dhcpcd tftp tftpd-server xinetd`. You're going to be working as root, so, if you need to launch a GUI tool such as gedit, use `sudo gedit` (as root).



## 03 Set up network

These examples use a machine with two network adaptors – one for connection to the outside world, and one to connect to the machines that need to boot from it. The second network card probably doesn't have an IP address assigned yet, so we'll set this via the GUI.



## 04 Configure second adaptor

Right-click on the network icon and select Edit Connections.... Now locate the second adaptor, click on Edit and select the IPv4 Settings tab. Change the method from Automatic (DHCP) to Manual. Add a static IP address for your adaptor. For example, if your first adaptor is on 10.0.1.1, adding the second adaptor with an address of 10.0.5.1 and a netmask of 255.255.255.0 will give you space to connect up to 255 machines to the boot server.

```
[root@killemika ~]# rpm -q dhcpcd tftp tftpd-server xinetd
dhcpcd-12.4.2.5-7.fc18
tftp-1.0.66-9.5.2-6.fc18
tftpd-server-1.0.66-9.5.2-6.fc18
xinetd-1.0.66-2.12.3.35-4.fc18

[root@killemika ~]# vim /etc/dhcpd.conf
```

## 05 Obtain dhcpd.conf

DHCP assigns IP addresses and starts the boot process on clients. Visit the official Fedora documentation ([tinyurl.com/luad-dhcp](http://tinyurl.com/luad-dhcp)) site to cut and paste an example DHCP configuration for a boot server. Load the existing file (`/etc/dhcp/dhcpd.conf`) into a text editor.

## 06 Modify dhcpd.conf

Modify the example `dhcpd.conf` to match your network. The subnet for our example would be changed to 10.0.5.0, and routers is the same address as your second network adaptor. Setting range dynamic-bootp to 10.0.5.50 10.0.5.100; gives space for 50 machines. The parameter next-server should be set to the same address as your second network adaptor. Change filename "linux-install/

```
dhcpd.conf (/etc/dhcp) - gedit
File Edit View Search Tools Documents Help
Open Save Undo Undo Redo Find Find Go To Go To
dhcpd.conf =
```

```
# DHCP Server Configuration file.
# see /usr/share/doc/dhcp*/dhcpd.conf.example
# see dhcpd.conf(5) man page

ddns-update-style none;
ignore client-updates;

subnet 10.0.5.0 netmask 255.255.255.0 {
    option routers 10.0.5.1;
    option subnet-mask 255.255.255.0;
    option domain-name "pxe-server.local.localdomain.net";
    option domain-name-servers 172.16.0.2;
    option ntp-servers 172.16.0.2;
    range dynamic-bootp 10.0.5.50 10.0.5.100;
    default-lease-time 21600;
    max-lease-time 43200;
}
allow booting;
allow bootp;
class "pxeclients" {
    match if substring(option vendor-class-identifier, 0, 9) = "PXEClient";
    next-server 172.16.0.5;
    filename "linux-install/pixelinux.0";
}
```

`pixelinux.0"; to filename "pixelinux.0"; Save the file and then start the server with systemctl start dhcpcd.service.`

```
tftp =
```

```
# default: off
# description: The tftp server serves files using the trivial file transfer \
# protocol. The tftp protocol is often used to boot diskless \
# workstations, download configuration files to network-aware printers,
# and to start the installation process for some operating systems.
service tftp
{
    socket_type             = dgram
    protocol                = udp
    wait                    = yes
    user                    = root
    server                  = /usr/sbin/in.tftpd
    server_args              = -s /var/lib/tftpboot
    disable                 = no
    per_source               = 11
    cps                     = 100 2
    flags                   = IPv4
}
```

## 07 Configure TFTP

TFTP is a basic file transfer protocol that the NIC firmware uses to fetch the bootloader. Load `/etc/xinetd.d/tftp` and change the line `disable = yes` so that it reads `disable = no`. TFTP is managed by xinetd, so start with `systemctl start xinetd.service`.

```
killemika@killemika:~$ yum install syslinux
File Edit View Search Terminal Help
syslinux           1.000        4.05-4.fc18          Fedora      1.0 MB
Transaction Summary
Install 1 Package
Total download size: 1.0 B
Installed size: 2.3 M
Is this ok [y/N]:
```

## 08 Obtain PXELINUX

PXELINUX is the Linux bootloader that works over Ethernet. To get it, install SYSLINUX with `yum install syslinux`. The file we need is `pxelinux.0`. Copy it to the TFTP folder with `cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/`. Type `mkdir pxeboot` to create the configuration directory.

```
default /var/lib/tftpboot/pxelinux.cfg - gedit
File Edit View Search Tools Documents Help
default.x

DEFAULT netinstall
LABEL netinstall
    kernel http://10.0.5.1/fedora/isolinux/vmlinuz
        append initrd=http://10.0.5.1/fedora/isolinux/initrd.img
        repo=http://10.0.5.1/fedora/
implicit 1
prompt 0

Plain Text Tab Width: 8 Line 9, Col 1 INS
```

## 09 Configure PXELINUX

Type `cd /var/lib/tftpboot/`. Make a directory with `mkdir pxelinux.cfg`. Within this directory, create a text file called `default`. Add the following lines: **DEFAULT netinstall**, **LABEL netinstall**, **kernel vmlinuz**, **append initrd=initrd.img** **repo=http://10.0.5.1/fedora/**, **implicit 1** and **prompt 1** so that it looks like the picture above. If you're feeling adventurous, try adding **prompt 0** so that clients won't wait for user confirmation before beginning the install. Be careful with that option!

```
killermike@localhost:/home/killermike
File Edit View Search Terminal Help
[root@localhost killermike]# mkdir /media/loop
[root@localhost killermike]# mkdir /media/netshare
[root@localhost killermike]# mount -o loop Fedora-18-i386-DVD.iso /media/loop
[root@localhost killermike]#
```

## 11 Extract the ISO image

Create two directories: `/var/www/fedora` and `/media/loop`. Copy the Fedora DVD ISO image to the current directory and type `mount -o loop -t iso9660 [path to ISO] /media/loop`. Use `rsync` to copy the files: `rsync -v -a -H /media/loop/ /media/var/www/fedora`.

```
killermike@localhost:/var/lib/tftpboot
File Edit View Search Terminal Help
killermike@localhost:[/var/lib/tftpboot] killermike@localhost:[/var/lib/tftpboot] killermike@localhost:[/var/lib/tftpboot]
[root@localhost tftpboot]# wget http://10.0.5.1/fedora/isolinux/vmlinuz
--2013-04-21 14:24:50-- http://10.0.5.1/fedora/isolinux/vmlinuz
Connecting to 10.0.5.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4895120 [text/plain]
Saving to: "vmlinuz"

2013-04-21 14:24:50 (41.5 MB/s) - "vmlinuz" saved [4895120/4895120]

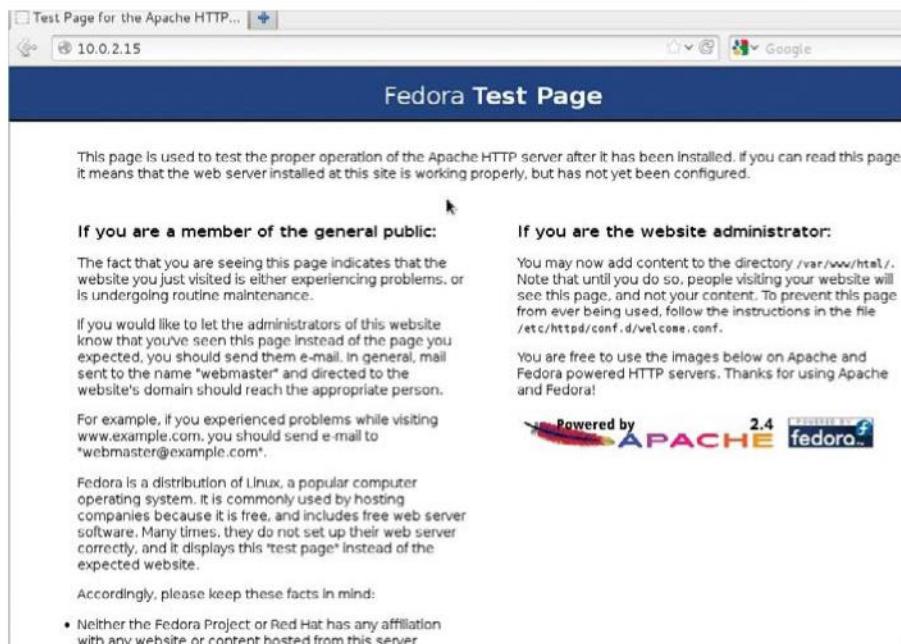
[root@localhost tftpboot]# wget http://10.0.5.1/fedora/isolinux/initrd.img
--2013-04-21 14:24:58-- http://10.0.5.1/fedora/isolinux/initrd.img
Connecting to 10.0.5.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 27496048 [text/plain]
Saving to: "initrd.img"

2013-04-21 14:25:00 (197 MB/s) - "initrd.img" saved [27496048/27496048]

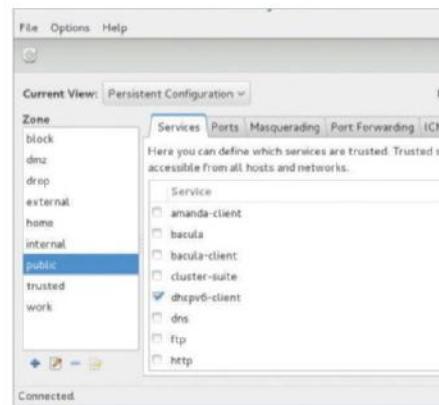
[root@localhost tftpboot]#
```

## 10 Configure web server

Add Apache 2 with the `yum install httpd` command, and start it with `systemctl start httpd.service`. Test that it is up and running by navigating a web browser to `http://10.0.5.1`. If everything's working, you should see the Apache startup page.



`wget http://10.0.5.1/fedora/isolinux/vmlinuz`. Copying the file like this makes a good test that the server is working. Now retrieve `initrd.img` from the same directory.

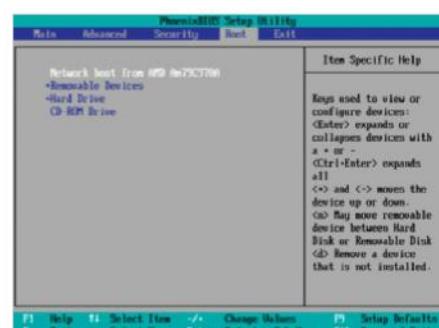


## 12 Copy vmlinuz and initrd.img

Enter the directory that TFTP can see with `cd /var/lib/tftpboot/`. Execute

**13 Firewall**

Open the Firewall configuration application. Select persistent configuration. Add http, https, tftp and tftpclient to the list of trusted services. Select Reload firewalld from the Options menu.



## 14 Ready the clients

Enter the BIOS setup screen of a client PC and make sure that the boot order specifies network booting as the priority. When carrying out the installation, you will disconnect the router/switch from the internet and connect it to the boot server instead.

## "A connected machine should now boot from the server"



### 15 Testing 1

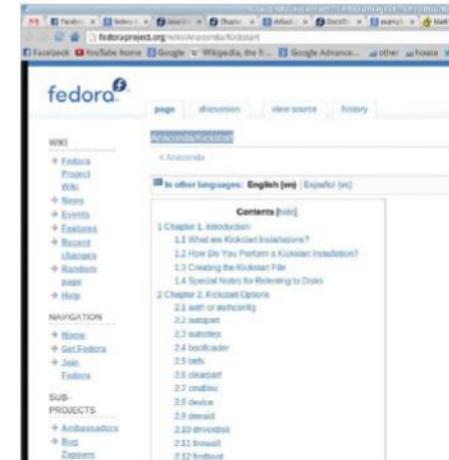
A connected machine should now boot from the server. If it doesn't work, there are some things you can try. Does the machine that is attempting to boot indicate that it has been assigned an IP address? If not, the problem lies with DHCPD on the server, so recheck /etc/dhcp/dhcpd.conf.

### 16 Testing 2

If the client tries but fails to load a file called pxelinux.0, it is communicating with DHCP, but TFTP may not be working. Try using the command `tftp 10.5.0.1 -c get pxelinux.0` on the server. If this retrieves the file, try executing it again on another machine. If the installer begins to boot, can find pxelinux.0, vmlinuz and initrd.img but stops at that point, try retrieving one of the files in /var/www/http/fedora/ manually by using the `wget 10.0.5.1/fedora/[name of file]` command.

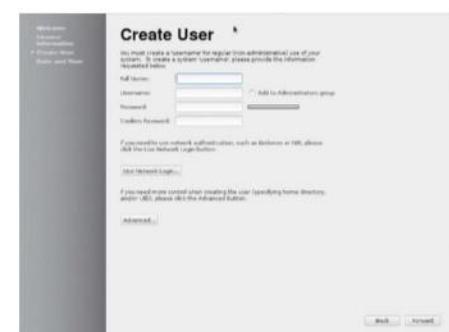
### 17 Make services permanent

Control Fedora services with `systemctl [command] [service]`. The main commands you'll need are `start`, `enable` to make permanent, and `restart` when you make configuration changes. This project requires running httpd.service, dhcpcd.service and xinetd.service.



### 18 Create Kickstart file

A Kickstart file supplies the installer with answers to avoid prompting the user. To begin, create a file called ks.cfg in `/var/www/html/fedora/`. Go to the official Fedora Anaconda/Kickstart page for a complete list of commands ([tinyurl.com/luad-kickstart](http://tinyurl.com/luad-kickstart)). When a Fedora system has been successfully installed, a (fully commented) Kickstart file is deposited in `/root/anaconda-ks.cfg`; this makes a good starting point for building your own. If you installed via the ISO, remove the line that sets install type to CDROM.



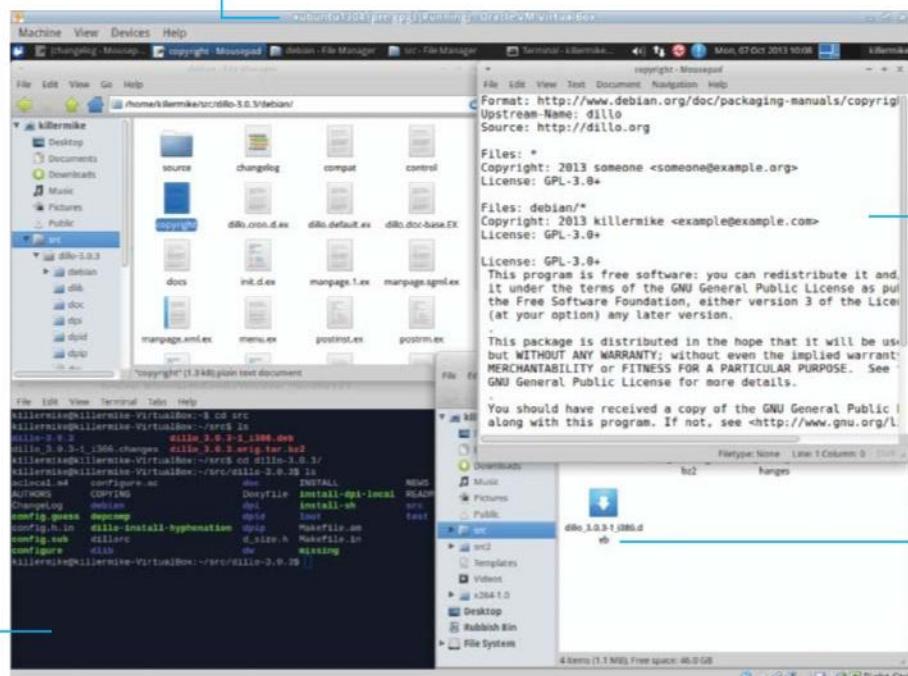
### 19 Configure Kickstart file

You must specify a root password using the `rootpw` command to avoid a prompt. You may want the installer to erase all partitions (or preserve some). The `upgrade` command causes the installer to upgrade the targets rather than carry out a fresh install.

Save your custom Kickstart file as `/var/www/html/fedora/ks.cfg` and then add `ks=http://10.0.5.1/fedora/ks.cfg` to the append line in your default file. By default, the finished target machines will begin in first-run mode and ask the user to specify details such as username and password on the first run.

The actual building of packages takes place from the command line, as does much of the setup

We recommend that you carry out this tutorial inside a virtual machine rather than on real hardware



Whether building RPMs or Debian packages, the configuration consists of editing some text files

The finished product, a DEB file that can be installed on Debian-derived distros

# Make your own DEB and RPM packages

We'll show you how to manufacture the two most common types of Linux package for software distribution so you can become your own package maintainer

We're going to take you through the process of making software packages for the two most popular packing systems, DEB and RPM. You can use these techniques to package your own software or even to become a package maintainer for software projects that you feel are being overlooked.

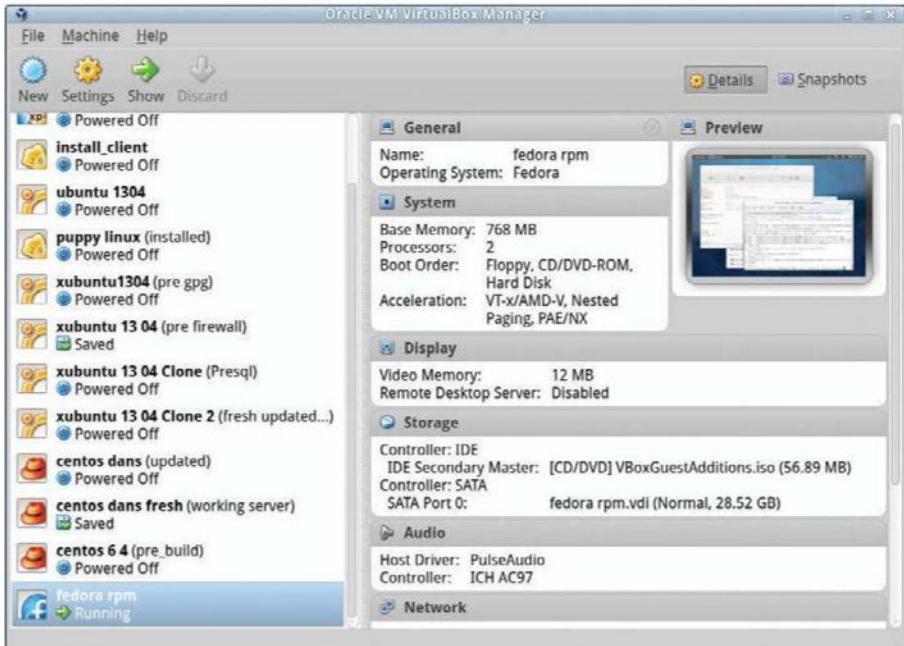
We'll start with a guide to building DEB (.deb) files for Debian-derived distributions – we're using Xubuntu as our base for that. Following that, we'll detail the methods needed for the creation of RPM packages for use on Red Hat-derived distributions, and we'll use Fedora for

that. You can often create a package on one distribution and then install it on a related one (Ubuntu>Debian, for example), but it might be worth testing it yourself, if this is crucial.

As for the software, we're going to use Dillo, a lightweight web browser, as an example package to build from source code. As is often the case when building from source, you may have to look around on the web for solutions if the build doesn't go as it should. For example, in the case of Dillo 3.0.3, we had to add 'LIBS=-lX11' to the front of the build commands to get it work, due to an oversight in the source code archive.

## Resources

Ubuntu & Fedora installation (or VM)



## 01 Employ a virtual machine

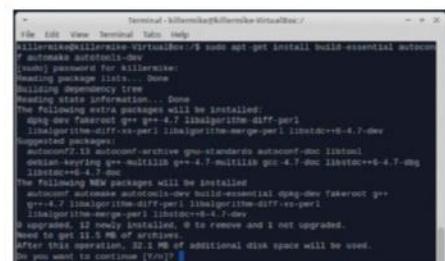
**01** Using a virtualiser such as VirtualBox or VMware is often the best approach to building packages for other systems. For one thing, it allows you to maintain a relatively clean, reference installation that is comparable to a setup that other people are likely to be running. This also means that you can keep a selection of target environments, using a different distributions. In addition, most virtualisation products allow the emulation of different architectures, and this can even extend to running a 64-bit OS on a 32-bit platform, although performance will suffer.



## 02 Starting from scratch

**02** If things go wrong, with Ubuntu or Fedora, it is perfectly safe to simply delete the source directory and start again. Note that the Debian tools do alter the source archive, so you'll have to start with a fresh copy.

## Part 1: Debian



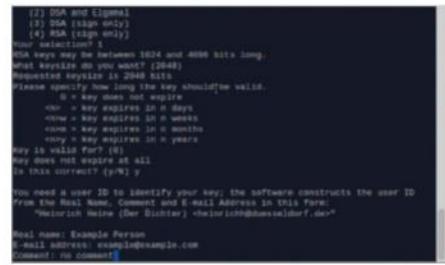
## 03 Install build environment

**03** We'll start by installing most of the tools that we need for to make software from source code. Type:

```
sudo apt-get install build-essential  
autoconf automake autotools-dev
```

Now we have to install tools that are used for handling DEB packages. Do this with the following command...

```
sudo apt-get install dh-make  
debhelper devscripts fakeroot xutils  
lintian pbuilder
```



## 04 Create a GPG key

If you haven't created or created a public GPG key in the past, you must create one now so you can sign packages. Start by typing `gpg --gen-key`. Accept the default settings, and fill in your details. Make a note of these, as we need an exact match later. Following this, type `ls ~/gnupg` to make sure the new key exists (it's `firstname_lastname.gpg`). Create a public key from this with:

```
gpg -a --output ~/gnupg/[your key].  
gpg --export '[your name]'. Import  
this with gpg --import ~/gnupg/[your  
key].gpg
```



## 05 Fetch package

**03** In this example, we're going to fetch and build the latest version of the Dillo web browser. Navigate to the Dillo website ([www.dillo.org](http://www.dillo.org)) and download the most recent .tar.bz tarball. Create a directory for source code with `mkdir ~/src` and move the archive into it.



## 06 Unpack the archive

**08** Unpack the archive with `tar -xjvf [archive name].tar.bz2`. Note the naming convention of the directory (package name-version) is crucial, and fortunately Dillo complies with this. It's also crucial that the source archive is one level above the source directory.

## 07 Add Debian compliance

**07** Move into the directory that we have just unpacked with `cd dh_make` is a script that takes care of adding the configuration file and directory structure that we need; it's part of the debhelper suite that we added earlier.

```
dh_make -e [your email address] -c  
licence -f ../[source archive]
```

```

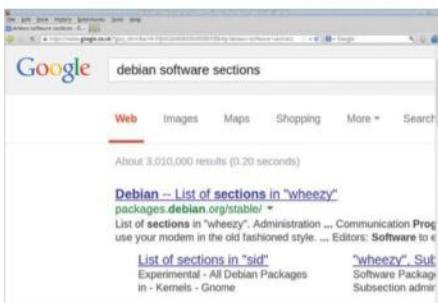
File Edit View Terminal Tabs Help
killermike@killermike-VirtualBox:~/src/dillo-3.0.3$ dpkg-depcHECK -d ./configure
Type of package: single binary, indep binary, multiple binary, library, kernel module, kernel patch?
[ctrl+D to exit]
Maintainer name : killermike
Email-Address   : test@example.com
Date           : 2013-04-04 06:10:48 +0100
Package Name   : dillo
version        : 3.0.3
License         : bsd
Upper Package   : dillo
MST 'enter' to confirm
none. Please edit the files in the debian/ subdirectory now, dillo
uses a configure script, so you probably don't have to edit the Makefiles.
killermike@killermike-VirtualBox:~/src/dillo-3.0.3$ 

```

In our example, the command line is:

```
dh_make -c gpl3 -e example@example.com -f ..//dillo-3.0.3.tar.bz2
```

When prompted, select single binary. The helper script should have created a directory called Debian within the source code directory.



## 08 Open the control file

Open the file control in the debian subdirectory in a text editor. Fill in the homepage section (Google for complete list of Debian software sections) and description fields of this file.

```

File Edit View Terminal Tabs Help
killermike@killermike-VirtualBox:~/src/dillo-3.0.3$ dpkg-depcHECK -d ./configure
Checking for pthread_create_in_ipthread... yes
Checking fontconfig presence... yes
Checking sysvapi.h usability... yes
Checking sysvapi.h presence... yes
Configuring for sysvapi.h... yes
Configuring for sysvapi.h... yes
Configuring for sysvapi.h... yes
Configuring for sysvapi.h... yes
Config status: creating Makefile
Config status: creating libdillo/Makefile
Config status: creating opengl/Makefile
Config status: creating x11proto/Makefile
Config status: creating x11proto-input/Makefile
Config status: creating x11proto-kb/Makefile
Config status: creating xorg/Makefile
Config status: creating xtrans/Makefile
Config status: creating xz/Makefile
Config status: creating doc/Makefile
Config status: creating dwm/Makefile
Config status: creating libdillo/Makefile
Config status: creating libxau/Makefile
Config status: creating libxcb/Makefile
Config status: creating src/ID/Makefile
Config status: creating test/Makefile
Config status: creating config.h
Config status: config.h unchanged

```

## 09 Discover dependencies

You can discover the dependencies needed to run the software in the package by moving into the source directory and typing `dpkg-depcHECK -d ./configure` into a terminal. This may produce errors that indicate a package needed in order to build the software is missing. You can discover these packages by typing `sudo apt-get build-dep [name of package]`, and this should help if there is some support for this software in the repository of the distribution. If not, you'll have to repeatedly

```

File Edit View Terminal Tabs Help
Terminal - killermike@killermike-VirtualBox: ~/src/dillo-3.0.3
checking for setsockopt... yes
checking for socklen_t... socklen_t
checking FLTK 1.3... no
configure: error: FLTK 1.3 required; fltk-config not found
Running strace failed (command line:
strace -e trace=open,execve -f -q -o /tmp/depchZSyRFw ./configure
killermike@killermike-VirtualBox:~/src/dillo-3.0.3$ sudo apt-get build-dep dillo
[sudo] password for killermike:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libpng12-dev' instead of 'libpng-dev'
The following NEW packages will be installed
libfltk-cairo1.3 libfltk-forms1.3 libfltk-gl1.3 libfltk-images1.3 libfltkk1.3-dev libglib2.0-dev libjpeg-turbo0-dev libjpeg8-dev libpcre3 libpcrecpp0 libpng12-dev libpthread-stubs0 libpthread-stubs0-dev libss libx11-dev libxau-dev libxcb1-dev libxdmcp-dev x11proto-core-dev x11proto-input-dev x11proto-kb-dev xorg-sgml-doctools xtrans-dev zlib1
The following packages will be upgraded:
libssl1.0.0
1 upgraded, 25 newly installed, 0 to remove and 178 not upgraded.
Need to get 8,891 kB of archives.
After this operation, 28.1 MB of additional disk space will be used.

```

**"If everything is set up correctly, we can finally build the DEB"**

run `dpkg-depcHECK -d ./configure` and add packages manually by typing `sudo apt-get install [name of package]`.

## 10 Add dependencies to the control file

When the command from the previous step has completed, it should present you with a list under the packages needed heading. Add this list of dependencies to the depends: section of the control file. Each item on the list must be separated by a comma and a space.

```

control - Mousepad
File Edit View Text Document Navigation Help
Format: http://www.debian.org/doc/packaging-manuals/copyright-format
Upstream-Name: dillo
Source: <url://example.com>

Files: *
Copyright: <years> <put author's name and email here>
           <years> <likewise for another author>
License: <special license>
<Put the license of the package here indented by 1 space>
<This follows the format of Description: lines in control file>
<Including paragraphs>

# If you want to use GPL v2 or later for the /debian/* files
# the following clauses, or change it to suit. Delete these
Files: debian/
Copyright: 2013 killermike <test@example.com>
License: GPL-2+
This package is free software; you can redistribute it and
it under the terms of the GNU General Public License as put
the Free Software Foundation; either version 2 of the License
(at your option) any later version.

This package is distributed in the hope that it will be useful
but WITHOUT ANY WARRANTY; without even the implied warranty
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

Filetype: None 

```

## 11 Edit the copyright file

Try to complete this step as comprehensively as you can, and don't skip it. Source: is usually the homepage of the project. Within the Files: \* section, replace the copyright information with the names of the authors of the project. You can see the required format for this by examining the Files: debian/\* section, which should have your details in it. You may have to do a bit of detective work to find the information you need. Look for files such as AUTHORS and COPYING within the source directory.

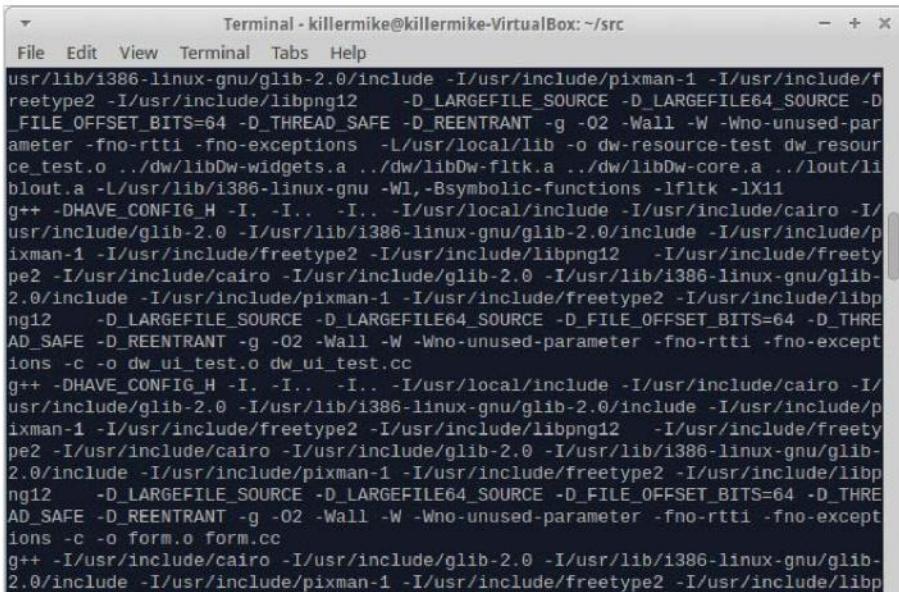
```

changelog - Mousepad
File Edit View Text Document Navigation Help
dillo (3.0.3-1) unstable; urgency=low
  * Initial release (Closes: #nnnn) <nnnn is the
  -- Example Person <example@example.com> Sun, 06
Filetype: Change

```

## 12 Edit the changelog file

Open the changelog file and make sure that the name and email address match those that you entered when creating your GPG key. Typically, the helper script may have added your username rather than your real name to the file. As with the copyright file, don't skip over

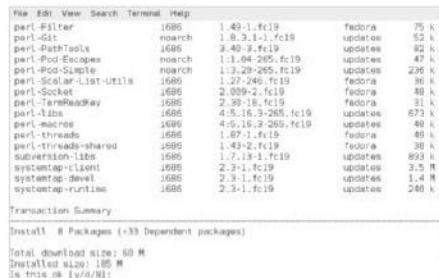


this section as doing so may halt the packaging process or lead to a non-compliant package.

## 13 Edit the changelog file

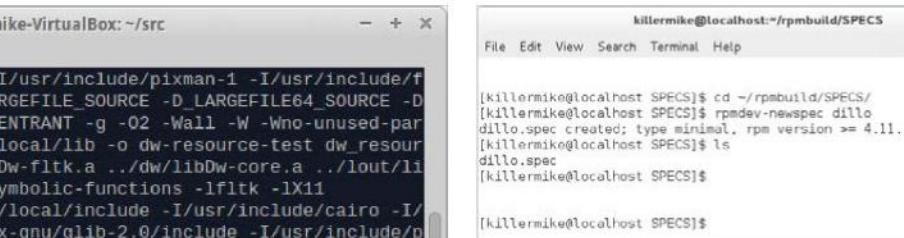
If everything is set up correctly, we can finally build the DEB. Move into the source directory and type `dpkg-buildpackage -b` to build the package, which is deposited in the `~/src/` directory. Example package by typing `dpkg -I [package]`. Run Lintian on it with `lintian [package]` to check for Debian policy compliance. Note that this tool is strict and it's up to you to decide if you can live with some minor non-compliance warnings. Finally, install it with `sudo dpkg -i [package]`.

## Part 2: Creating RPMs with Fedora



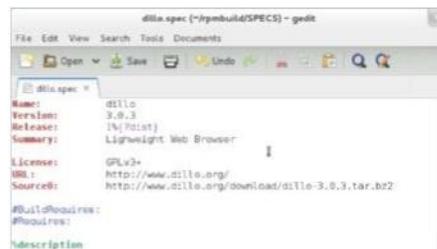
## 1/ Open the control file

**14** Become root by typing `su`. Begin with `yum groupinstall "Development Tools"`, and follow this up with `yum install gcc-c++ fedora-packager`. Type `usermod -a -G mock <your username>` to add your user to the mock group.



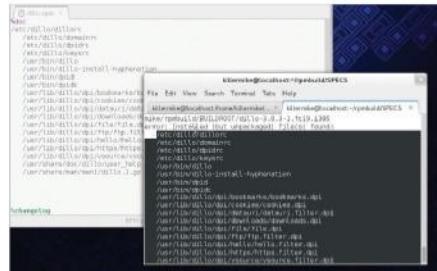
17 Create .spec file

Red Hat derived distros such as Fedora use .spec files to specify the build process. Move into the directory that contains these files with `cd ~/rpmbuild/SPECS/` and create a blank .spec file by typing `rpmddev-newspec dillo`.



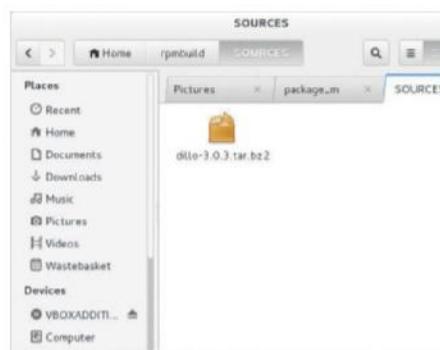
## 18 Edit .spec file

**10** Type `gedit` `dillo.spec`. Fill in the Version, Summary and Licence (GPLv3+ in this case) fields. URL is the project homepage; Source0 is the URL of the source code there. Comment out `BuildRequires` and `Requires`. Add a full description in the `%description` area.



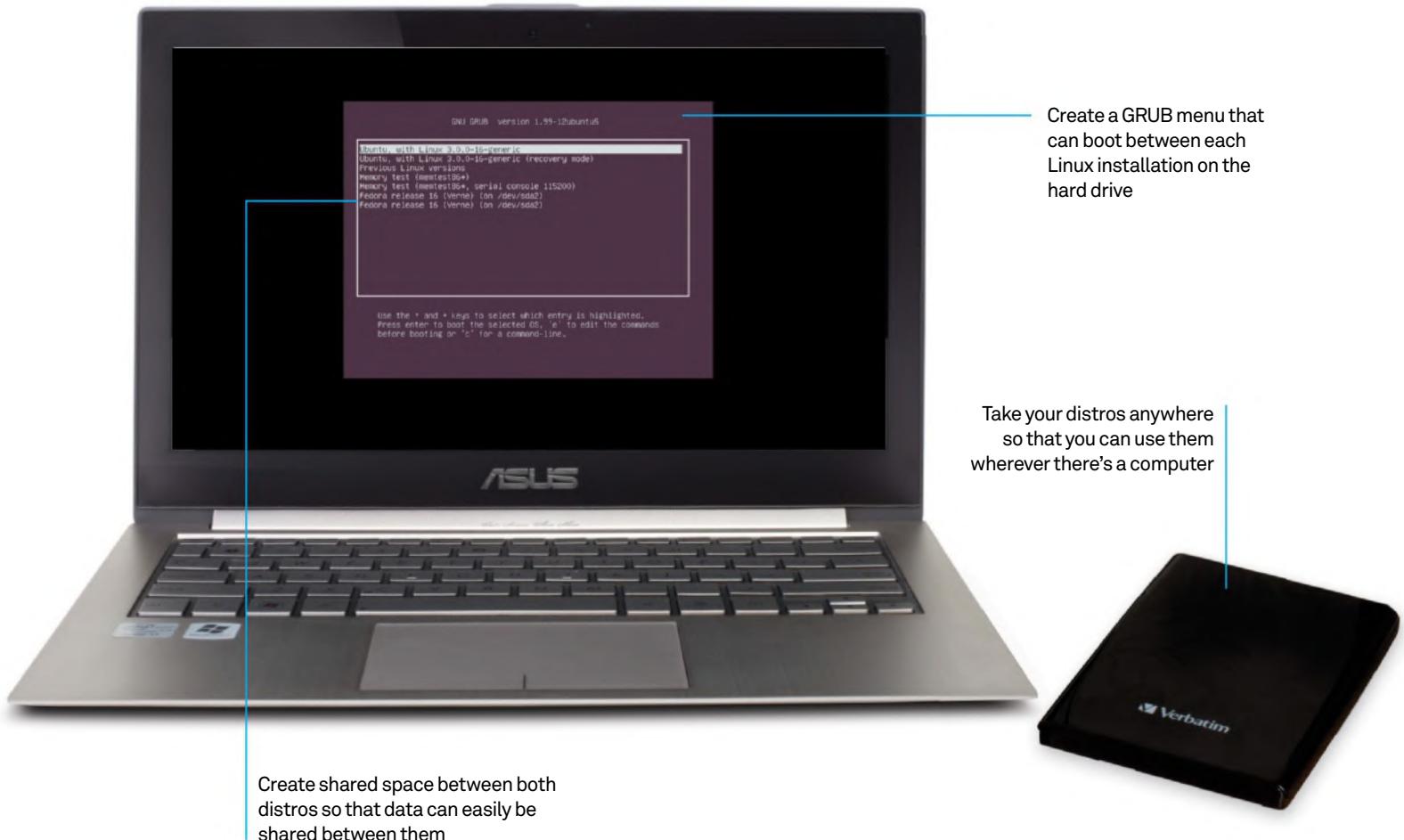
## 10 Build source code

If the package is supported at all in the system, run `yum-builddep [name of package]`. Otherwise, you'll have to repeat the build command to generate errors or search the documentation in the source archive. In the SPEC directory, type `rpmbuild -ba [name of package].spec`. If this build fails and reports about extra, unpackaged files, cut and paste this list of files into the %files section of the .spec file and repeat the build command. The package is now in the RPMS directory. Type `rpm -ivh [package]` to install it. Type `rpm -qa | grep [package]` to make sure it is installed.



## 16 Fetch the archive and move it

**b** Download Dillo from the Dillo website and move the archive into the proper directory by typing `mv [name of archive] ~/rpmbuild/SOURCES.`



# Dual-boot from an external hard drive

Want to carry around a multi-booting hard drive you can connect to any computer? Then this easy-to-follow tutorial is exactly what you need

## Resources

A USB-powered external hard drive with at least 75GB of space

**Carrying around a USB stick with a preloaded live distro is very useful.** It can be used as an emergency distro (for a PC experiencing some troubles), diagnosis tools, a portable distro for yourself and much more. However, when it comes to using it as a portable distro, there are some limitations. Not all live distros allow you to install extra software permanently, and even then they may have a limited repository of software that can be installed in a live environment.

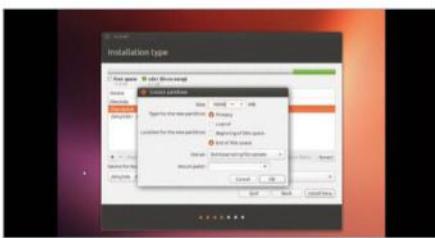
This isn't a problem if you have enough storage space, though, such as on a portable, USB-powered external hard drive. You can install distros to them much as you would on a normal hard drive. However, there are some changes you'll need to make if you plan to dual-boot from this hard drive on any machine.

Here we show you how to partition, install and run two distros from an external drive. We'll use Fedora and Ubuntu in our example, as they are two of the most common distros.



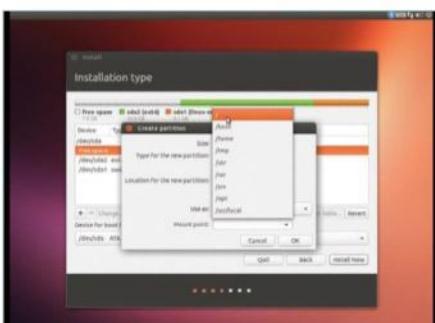
## 01 Swap partition

In our example, we'll start with Ubuntu. Plug in the USB drive and boot into Ubuntu. Make sure you've saved any important data on the external drive. The first thing you'll need to do is create a swap partition – make it 4GB and put it towards the end of the hard drive.



## 02 Shared space

You can create some shared space for both distros to use, either independently or as a shared home directory. We want to save 10-15GB for each distro, so keep that in mind while creating it. Use ext3 or ext4 as the file system, or NTFS if you want it to be cross-platform.

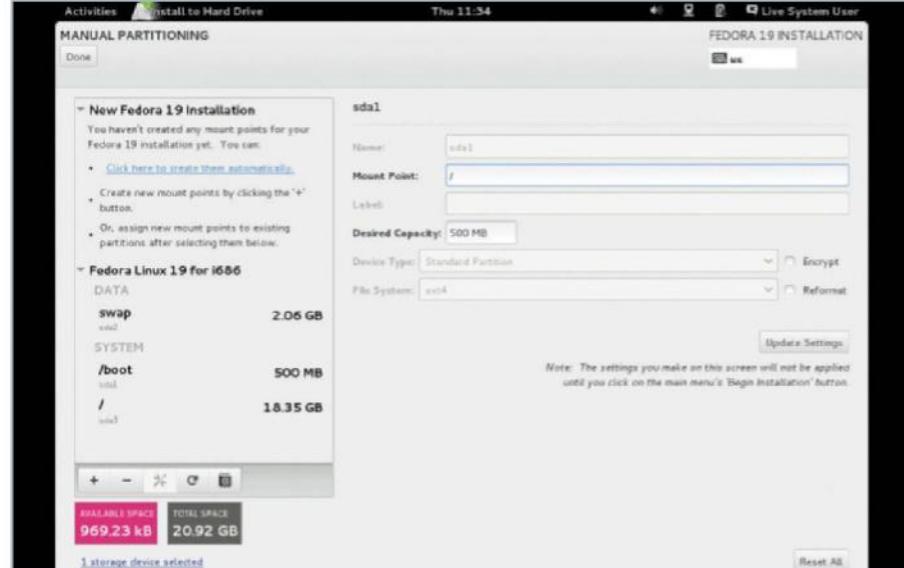


## 03 Ubuntu root

Create the 10-15GB space for Ubuntu and set it to be root by selecting '/' in the menu. It'll automatically mount our swap partition when booting into Ubuntu/Fedora in the future. Carry on with the installation instructions.

## 04 Fedora root

Put in the Fedora disc with the USB hard drive still plugged in and boot up. Choose the external drive from the list of disks in



Installation Destination, create a custom partition in the space remaining as ext3 or 4, and set the mount point to '/'. Install as normal.

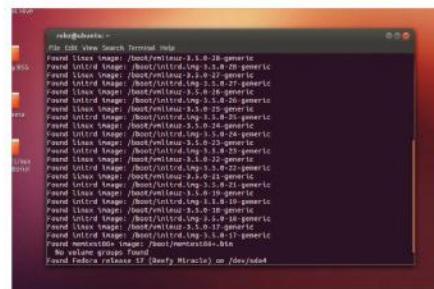
## 05 GRUB reinstall

Boot back into the Ubuntu live disc and make a note of what `fdisk -l` labels the Ubuntu boot partition on the external drive as. Mount it with:

```
$ sudo mount /dev/sdXY /mnt
```

And then reinstall GRUB 2 from Ubuntu with:

```
$ sudo grub-install --root-directory=/mnt/ /dev/sda
```



## 06 GRUB update

After rebooting, you'll be able to get back into Ubuntu on the external drive. Once there, mount the Fedora partition however you wish and run:

```
$ sudo update-grub
```

It will automatically detect the Fedora install

and update the boot menu next time you boot from the hard drive.

## 07 Find fstab

Enter the command:

```
$ ls -l /dev/disk/by-uuid
```

...to find out the UUID of the partition for your shared storage on the hard drive. This should be called the same on both Ubuntu and Fedora. Go to the terminal and open fstab with your favourite text editor like so:

```
$ sudo nano /etc/fstab
```

## 08 Use fstab

You'll need to add a new entry so that your shared space mounts every time you boot into one of the external drive distros. Enter into both fstabs something like:

```
UUID=XXXXX [mount point] ext4  
errors=remount-ro 0 2
```

You can change ext4 to whatever other file system you used.

## 09 Go portable

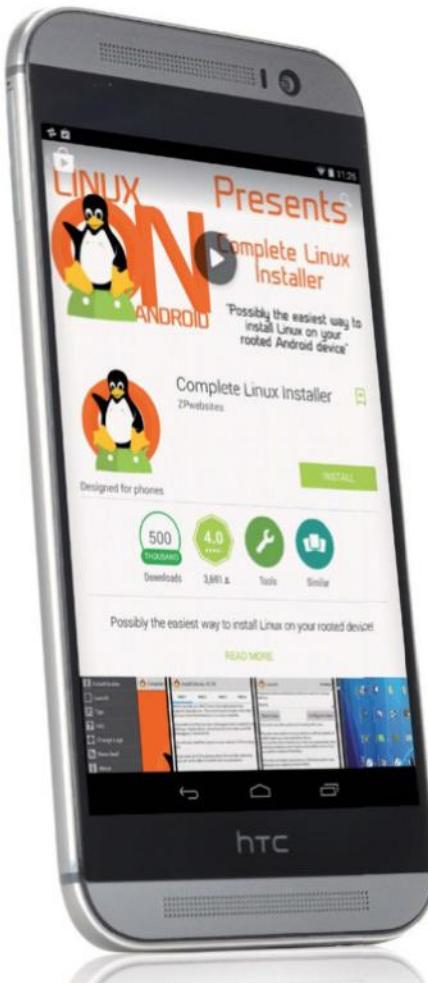
You should now be fully set up and ready to take your portable Linux distributions everywhere, with a shared space between them, and the ability to boot between either of them without having to guess which partition is which.

# Run Linux on your Android device



Want an ultra-portable version of Linux? Then put it on your phone





**Above** Sadly, we experienced a few problems with the Complete Linux Installer app

If you can't wait for the launch of the official Ubuntu smartphones (the first models are supposedly due early this year), don't want to shell out for a new phone anyhow, or would prefer to use a different version of Linux on a portable device, there is an alternative. It's possible to run a variety of popular Linux distros on a standard Android smartphone or tablet – everything from a simple BusyBox toolset right up to a full distribution with a desktop environment. You don't even need to root your phone for some of the methods that we explore in this feature.

The advantages of running Linux on an Android device are manifold. As well as being able to SSH into other computers, you'll have access to all your favourite Linux tools and you can also run a desktop GUI with most methods. The possibilities are endless. You could potentially even turn your Android device into a LAMP server to run web apps! So, if you've got an ageing Android phone or tablet kicking around, why not give it a try?

### No rooting required

As mentioned, some solutions for running Linux on an Android phone don't even require you to root the device to circumvent Android's security features and gain superuser privileges – although

## Rooting questions

The ins and outs of rooting an Android device

### What is rooting?

In short, it is gaining root access to control some of the Android system's secure features. This involves unlocking the device's bootloader, replacing the recovery system with a custom version and installing a superuser app.

### What benefits does it offer?

Advantages include the ability to modify every aspect of the Android system, install custom ROMs, overclock the CPU, install extra apps, remove OEM bloatware and, of course, run a full-blown Linux distro.

### How difficult is it to do?

The process differs for every Android device, but often entails connecting it to a computer. For this, the device must have USB Debugging enabled in Settings>Developer Options, which appears when you tap the Build Number seven times on the About Phone/Tablet screen. If the manual command-line method using the ADB and Fastboot tools is too daunting, there are

numerous desktop utilities available to simplify the process. There's even an Android app, Towelroot (<https://towelroot.com>), that can root many devices with a single button tap.

### What are the downsides?

You'll almost certainly void your warranty and there is a low risk of soft-bricking your device (ie it won't boot up properly), but you should be able to restore it to working order using a connected computer with the adb and fastboot tools to re-flash the original Android ROM. Also, unlocking the device's bootloader will perform a factory reset that wipes the device's data and downloaded apps, so you'll want to be sure they're backed up first. While, by default, app data is backed up to your Google account in the cloud, it's not exactly transparent and you'll need to re-download the apps afterwards, so you might want to do a further computer backup. A good way to do this on a non-rooted phone is with ClockworkMod's Helium Android app ([goo.gl/Dq3o4F](http://goo.gl/Dq3o4F)) and desktop client.

## KBOX2 BusyBox

An easy-to-install BusyBox with some standard Linux tools

The KBOX2 project ([kevinboone.net/kbox2.html](http://kevinboone.net/kbox2.html)) works by constructing a minimal Linux root file system within the private data area of the hosting terminal emulator app, so you'll need to download one to use it – try Android Terminal Emulator by Jack Palevich ([goo.gl/p8RPH2](http://goo.gl/p8RPH2)). While you can install KBOX2 manually from that app's command line, it's easier to use the OneBox Package Manager app ([goo.gl/3960TO](http://goo.gl/3960TO)) – you'll need to buy a companion app (£2.03/\$3.10). Just follow the in-app directions and it'll run the setup script for KBOX2 within the terminal emulator. Now go to the latter and enter: /data/data/jackpal.androidterm/kbox2/bin/kbox\_shell. The prompt should change to /home/kbox \$. You can then download the packages you need from the KBOX2 site (via the device's web browser) and install them with: dpkg -i /sdcard/Download/{package}.deb. Compatible packages include Perl, Dropbear (SSH support), GCC, Vim and rsync.



## Launch a distro with GNURoot

The GNURoot app makes it easy to run several distros in a terminal emulator...

**01** Search for GNURoot in the Play Store on your Android device. Install the main GNURoot app, then the helper app of whichever distro you want to run with it.

**02** Open the GNURoot app and choose the distro for the helper app you're using from the top drop-down menu. Then tap the Create New Rootfs button to unpack it – this can take a few minutes.

**03** Once done, select that distro from the second drop-down and tick the 'Launch as Fake Root' box (so you can use apt-get and other root commands), then tap Launch Rootfs.

**04** It may take a while to start the first time, resulting in a black screen before a familiar command-line terminal appears. You are now ready to use the downloaded distro.

**05** First, to ensure everything's up to date, use the commands `apt-get update` and `apt-get upgrade` in Wheezy. If using Fedora, instead use `yum update`.



**“To run programs requiring windows, you’ll need a GUI. Luckily, GNURoot does offer a way to implement one”**

## ARM-less devices

### Running Linux on x86-based Androids

For this article, we've focused on installing Linux on ARM-based devices. However, some Android devices, such as the Samsung Galaxy Tab 3, are based on the x86 architecture. So, since nearly all Linux distros are based around x86, in theory they should easily run on these devices – whether using a live USB stick (via USB2Go) or a full install. You may need to tweak settings to get some features working, such as the virtual keyboard and screen rotation. Ubuntu and openSUSE (with its TabletPC pattern) seem to be the best options.



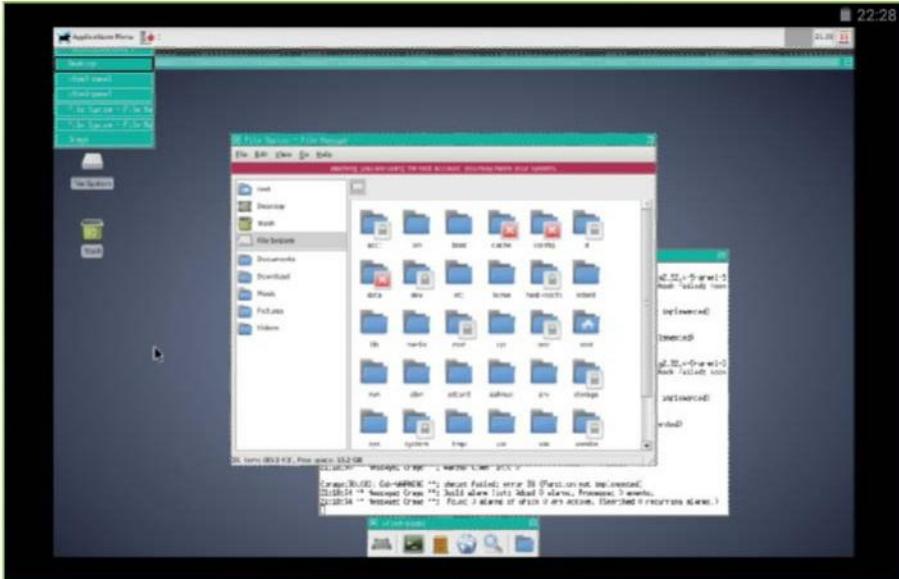
devices by using a ptrace container implemented via the PRoot utility. No need to worry about any technical jargon, however, since the app does all the nitty-gritty for you and is very easy to use. After downloading and installing the main GNURoot app, you then need to download and install one of the 'helper' apps available for different ARM-based distros: (Debian) Wheezy ([goo.gl/nP09Ex](http://goo.gl/nP09Ex)), Fedora ([goo.gl/ZGZj4N](http://goo.gl/ZGZj4N)), Aboriginal (a lightweight BusyBox variant, [goo.gl/7K4Dec](http://goo.gl/7K4Dec)) and Gentoo ([goo.gl/T4Y8Ja](http://goo.gl/T4Y8Ja)). There's also an x86 version of Wheezy available if you have an x86-based device ([goo.gl/6qr9ot](http://goo.gl/6qr9ot)). The downloaded distro can then be unpacked and launched using the simple menu system in the main GNURoot app, which does all the hard work and eventually places you in a command-line terminal emulator (see 'Launch a distro with GNURoot' boxout above for more details). You're then all set to go and should be able to install any packages from the distro's repo in the usual fashion, using the relevant package manager, such as apt-get or yum. You can create extra terminal windows by tapping the '+' button

at the top, then switch between them via the top-left drop-down menu. The top-right (three-dots) icon also brings up a menu for various settings, including font size (you might well want to increase it to see the text more easily on a small screen) and a list of special keys accessed via various volume button combinations.

### Getting graphical

While useful for running various Linux tools and utilities, one obvious limitation of GNURoot's four main distro options is that they only run via the command line. To run programs requiring windows, you'll need a GUI. Fortunately, GNURoot does offer a way to implement one, using the helper app called 'GNURoot Wheezy X (xterms)', which launches a VNC server for this purpose ([goo.gl/P0ezIJ](http://goo.gl/P0ezIJ)). To see the GUI, you'll also need to download one of the many VNC clients available in the Play Store – we used VNCViewer.

The WheezyX distro launches in a terminal window. After updating and upgrading, as before, note the address of the VNC server at the top – it should be 'localhost:1' the first time. You can then open the VNC Viewer app and point it to this address, entering the password as 'password'. A virtual desktop will then launch – note that there's no proper desktop environment by default; just an xterm terminal window for Wheezy. By swiping around the screen, you can move the mouse pointer onto this window and



tap to select it. You can use the keyboard icon in the pop-up top toolbar to start typing into the terminal, and use the handy row of special keys above the on-screen keyboard for things like Ctrl, Alt, the cursor arrows and function keys.

From this terminal window you can install and launch programs to run within windows on the desktop. Using the mouse pointer, windows can be dragged around and resized. While a bit fiddly, the system works pretty well, although you may get the odd error and applications may well be missing audio (a common problem when using VNC viewers). You can also install a desktop environment via the terminal. We managed to get Xfce working by installing it with: **apt-get install xfce4**. We then launched it from the VNC Viewer terminal with: **startxfce4**.

For some reason, the main desktop launched inside a window of its own, but could be dragged into position to fill the screen. There's an application menu at the top for launching programs, although some items don't work by default. So you may need to tinker around with the system to get it working better for you.

### Full installation

If you want to make full use of Linux on your Android device, the best solutions require rooting it and unlocking its bootloader (see 'Rooting questions' boxout on the previous page). Whichever way you do it, this is a major step as it will void your warranty and also runs a risk of 'soft-bricking' the device – although it can be made to work again if that happens. Another drawback is that unlocking the bootloader will factory-reset

your phone and erase all its apps and data, so ensure that you make a backup beforehand.

Once you've rooted your phone and unlocked the bootloader, you are able to install and run a compatible Linux distro within a chroot environment on the device. The easiest way to do this is by using one of the installer apps available in the Play Store. We tested out Complete Linux Installer ([goo.gl/Js6wnL](http://goo.gl/Js6wnL)) and Linux Deploy ([goo.gl/cdp6FO](http://goo.gl/cdp6FO)). Both of them enable you to run a run a selection of popular Linux distros within a chroot environment on the Android device – it's not a virtual machine since they run directly on the ARM architecture. You can then access the running distro from an SSH/terminal app or, if using a GUI, a VNC viewer.

While, in theory, the Complete Linux Installer's built-in guides and easy-to-use interface make it possible to install and launch a distro – choose from various flavours and sizes of Ubuntu, Debian, ArchLinux, Kali Linux, Fedora and openSUSE – with just a few button taps, we experienced a few problems getting most of them to work. Still, the team behind it are currently alpha-testing an improved version three, so you might want to give that a try when it's ready.

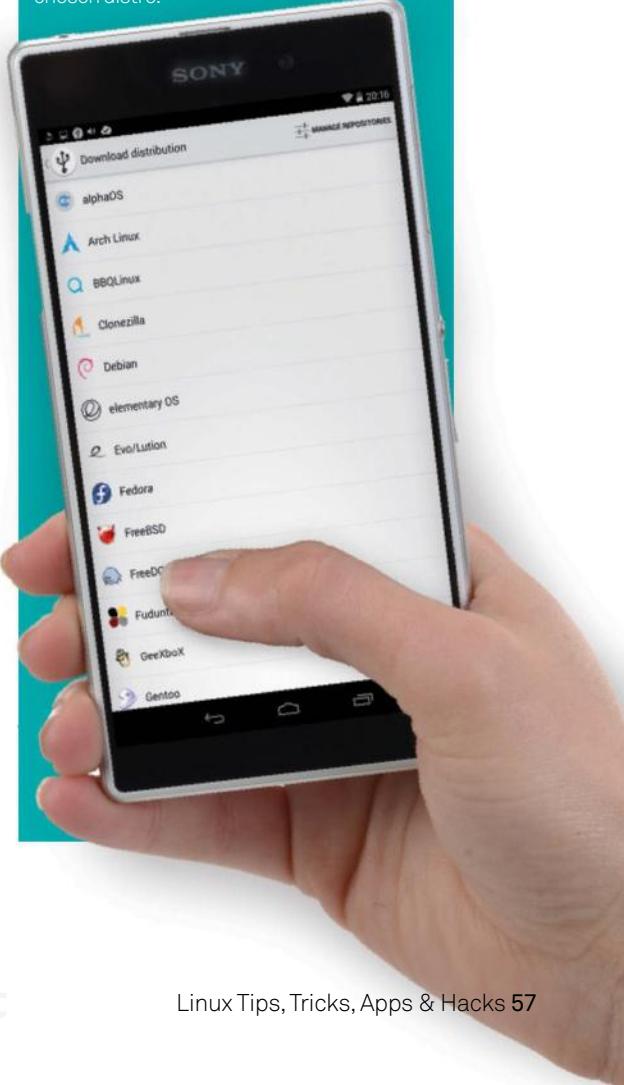
However, here we'll focus on Linux Deploy, which proved more reliable. Launching the app takes you to a simple black screen with your device's IP address and some options at the top (see 'Run a distro with Linux Deploy' boxout above). Tapping the down-arrow icon takes you to the settings screen where you can choose a distro to install: Debian, Ubuntu, Arch Linux, Fedora, openSUSE, Kali Linux and Gentoo

**Left** Here's Debian Wheezy for GNURoot running the Xfce desktop environment

## Boot a PC from your Android

### Use DriveDroid to boot into Linux

If you prefer to use Linux on a PC then a live CD or USB stick always comes in handy for booting it up wherever you go. Even better, the DriveDroid app ([bit.ly/1ttEmEO](http://bit.ly/1ttEmEO)) will enable you to do so from your (rooted) Android phone, on which you can carry several distros. Just download and enable an ISO or IMG file in the app, connect via USB to the PC, then reboot the latter (having placed your device at the top of the booting order in its BIOS). It'll then boot into the chosen distro.



## Run a distro with Linux Deploy

Easily install the distro of your choice on a rooted Android device

**01** Launch Linux Deploy and tap the down arrow icon at the top to access the settings. First, tap Distribution to choose a distro. Then tap 'Distribution suite' to choose a version, and Architecture if you want to change that too.

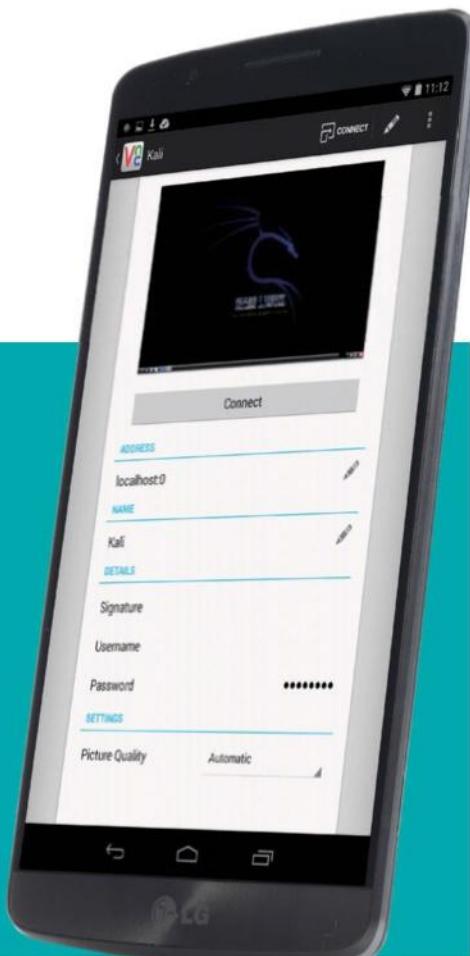
**02** Scroll down the settings and tap 'Desktop environment' to choose one. SSH and GUI are enabled by default. Tap GUI settings to alter the width and height settings to suit your screen (you can always change them later).

**03** When all the settings are made, tap Install at the top to begin the installation. After making the .img file, all the

installation processes will be detailed on screen. When you see 'end: install', the distro is ready to launch. Tap Start.

**04** To access the distro from a terminal, launch your SSH app and (if using default settings) log into android@<your device IP address> with the 'changeme' password. You can now enter standard Linux commands from the terminal.

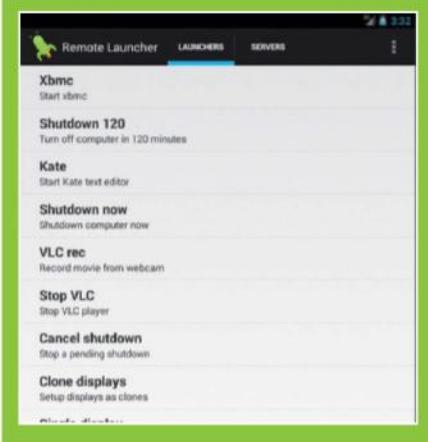
**05** To use a GUI, launch your VNC viewer app. Use 'localhost:0' as the address, port 5900 and the 'changeme' default password. Once connected, the desktop will appear – if not sized correctly, change the GUI settings in Linux Deploy, then Stop and Start.



## Remote control

Control your PC from your phone

Another way of using your Android device with a PC is as a remote control. The Linux Remote Control web app enables you to access any Linux box on the same network with the desktop client installed. You can control things like music and video playback, the mouse pointer, restart and shut down, and send custom commands to run on it. Alternatively, the Remote Launcher Android app can launch applications on any PC running the Java Runtime Environment. While not as versatile as SSH or VNC, both apps are easy to use.



are available, plus a RootFS option. Further settings enable you to change the version and architecture (useful if your device doesn't have an ARMv7 CPU). You can also choose a desktop environment (LXDE, Xfce, GNOME, etc) and alter VNC display settings to suit your screen resolution. When ready, hit the Install option at the top and Linux Deploy will begin installation – by default, to an IMG file. Naturally, this may take some time – usually 30 to 60 minutes, depending on the distro and your Wi-Fi connection speed (don't do it over 3G!) – with all the processes shown on screen. You'll know it's finished when you see the 'end: install' line. So long as it didn't fail, you're ready for launch.

Hit the Start button to boot the Linux instance; SSH and VNC servers will be launched by default. Note that you can't enter any commands from Linux Deploy itself – you need to connect from an SSH app, such as VX ConnectBot or JuiceSSH. You can then enter standard Linux commands from the terminal.

To access the GUI and desktop environment, you'll need to use a VNC app such as VNC Viewer. If the desktop has the wrong dimensions, just go and change the GUI settings in Linux Deploy and restart the distro from there.

While we enjoyed considerable success using Linux Deploy, we did encounter a few teething troubles. If the distro installation fails for some reason, in most cases simply hitting Stop and then reinstalling fixes the problem. We also sometimes got a 'failed to truncate' error when Linux Deploy was making an IMG file at the start

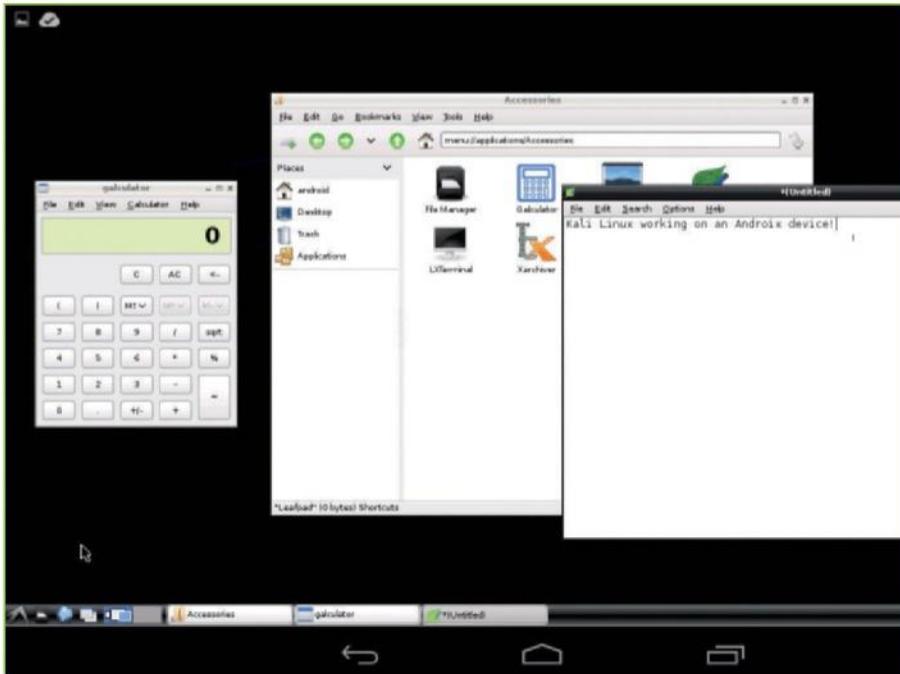
of an installation; we found that reducing the image size solved this, though it's not ideal as it reduces the space for the distro. Indeed, a limitation is that Android's FAT32 file system for SD cards has a maximum file size of 4GB, so you can't install a distro .img file greater than this. A solution is to create two partitions on the card – FAT32 and ext2/3/4 – and then install the distro to the latter (either as a file or partition); if your device only has internal storage, you may be able to use a workaround involving a USB2Go cable and pen drive.

## Magic touch

While it's great to be able to run a desktop distro on your phone, navigating it with a mouse pointer and virtual buttons can be fiddly. So you might want to give Ubuntu Touch a try. Not to be confused with Ubuntu for Android (which is used via a monitor and physical keyboard), Touch puts a special touch-screen version of Ubuntu on your device. Unlike the other methods for running distros, it doesn't run in a chroot environment within Android – instead, you install it as a standalone OS in the recovery partition of your device and you can then dual-boot it with Android via the fastboot menu.

While officially, Ubuntu Touch only supports certain Samsung Galaxy and Google Nexus devices, it can be made to run on others. See the website for more details: [wiki.ubuntu.com/Touch/Devices](http://wiki.ubuntu.com/Touch/Devices).

Installation is surprisingly easy if you've already rooted your device and unlocked its bootloader.



**Above** Like the other Linux Deploy distros, Kali comes with a basic set of desktop applications

**“While it’s great to be able to run a desktop distro on your phone, navigating it with a mouse pointer and virtual buttons can be fiddly”**

Simply download the Ubuntu installer ZIP file ([bit.ly/1oZC2xJ](http://bit.ly/1oZC2xJ)) and move it to the root of your SD card. Then put your device into fastboot mode: turn it off, then hold the power and volume-down buttons until you see the green robot screen with a large Start option. Use the volume-down button to switch to Recovery mode and then press the power button.

Now use the menus to install the Ubuntu installer ZIP from your SD card (in the /0 directory). Reboot the device and you should find the Ubuntu Dual Boot app in your Android apps menu. You’ll need to ensure you can run apps from unknown sources (under Settings>Security). Now tap the app icon, then ‘Choose which channel to install’ to select a version of Ubuntu. You’ll need to wait quite a while for the app to download and be installed.

When it’s finished, tap the ‘Reboot to Ubuntu’ button to launch Ubuntu Touch on your device. The gesture-based navigation system is easy

to use: swipe from the left edge to open the launcher, from the right for the app switcher, from the top for a quick settings panel, and from the bottom for extra options. Linux commands can be entered via the Terminal app. Note that since Touch is still in beta, there are a few issues, depending on the device, but it’s usable and gives you a good preview of how the finished system will work.

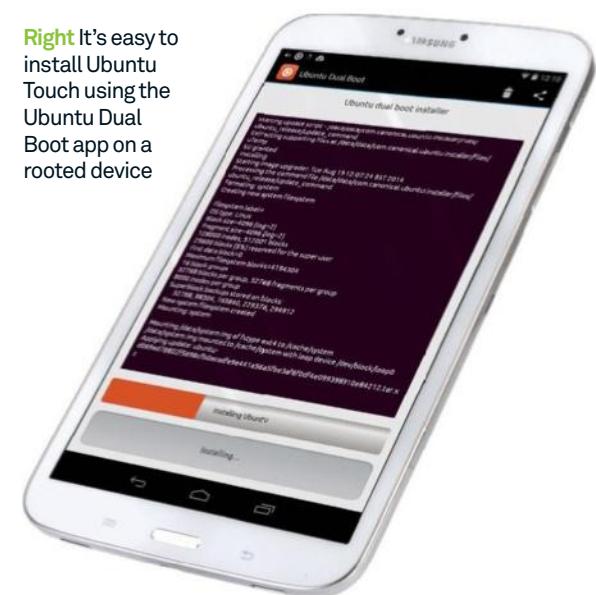
To return to Android, simply power down, then hold the power button to launch it. To restart Touch from Android, open the Ubuntu Dual Boot app and hit the reboot button. You can also uninstall it from the settings menu here.

As we have demonstrated, there are many ways to install Linux on your Android device, each with its own pros and cons. There are also many things you can do with it once installed – see the ‘Why do it?’ boxout to the right for some ideas. So, why not get Linux onto your phone and take it with you wherever you go?

## Why do it?

The key benefits of running Linux on Android

- It’s portable and convenient. No need to carry around a Linux laptop when you can run Linux on your Android phone instead.
- Access the command line from a terminal emulator and use all of your favourite Linux tools.
- Or use a VNC viewer to run a desktop GUI and windowed applications.
- Gain access to thousands of Linux applications. Many of these are more powerful and more customisable than their Android equivalents.
- It’s particularly useful for power users and software developers; in order to build ARM software, and have real git-annex repositories, for instance.
- You could even run a LAMP server from your device, to run web apps. It’s relatively straightforward to configure.
- A Linux-running Android device could be used in hardware projects, such as robots – either in place of a Raspberry Pi or as a remote control unit.
- Gain all the benefits of running Linux without harming your standard Android environment, giving you the best of both worlds on one portable device.

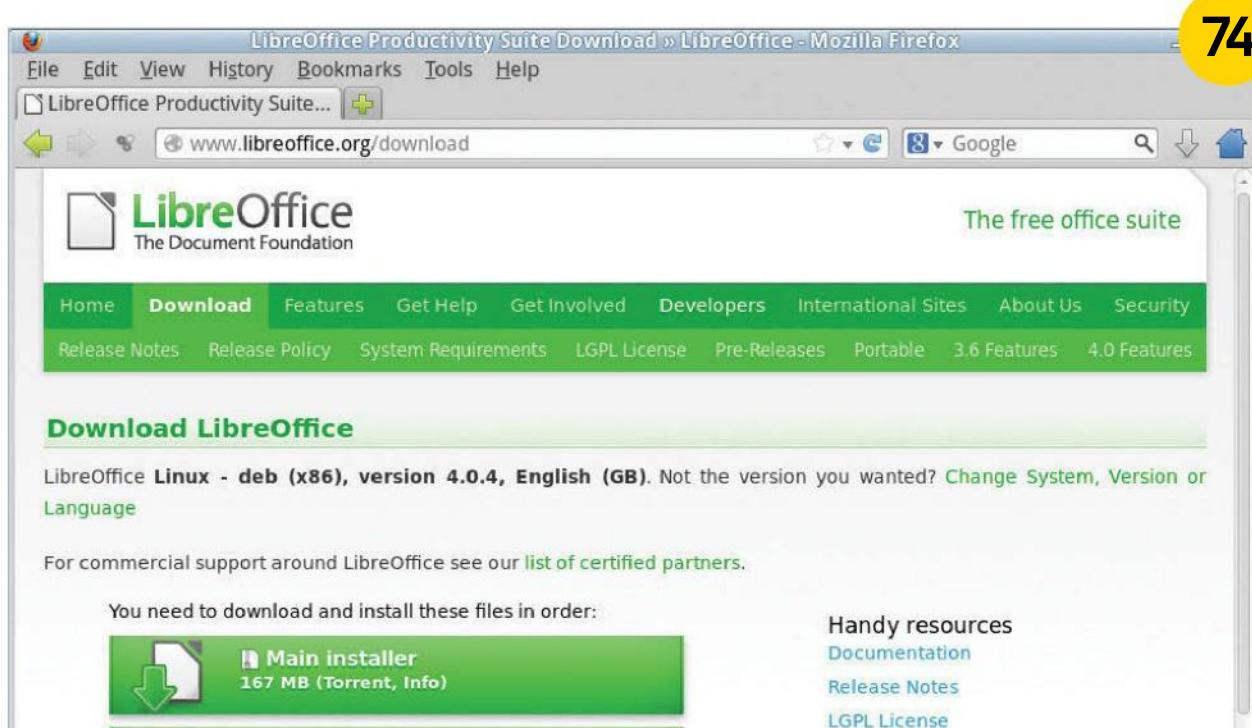


# Tricks

## Unlock the true power of open source

- 62** Network wirelessly with **wicd**  
Get your wireless network up and running
- 66** Manage your system with **Webmin**  
A great GUI front-end for system configuration
- 70** Synchronise your files with **Unison**  
Use this command-line tool to sync files between computers
- 74** Make a small business database with **LibreOffice**  
Create a simple, form-based database

- 78** Create and save data with a **MongoDB database**  
Forget about joins and SQL and try NoSQL databases
- 82** Maintain and manage all of your machines with **Puppet**  
Keep them in a consistent and workable state
- 86** Visualise directory structures with **Graphviz**  
Make large directory structures practical



**90 Edit videos in Kdenlive**

Create great-looking videos with open source software

**92 Build your own private cloud with ownCloud**

Set up your own file management system in the cloud

**96 Design exciting presentations with Hovercraft**

Use impress.js and reStructuredText to create presentations

**100 Set up a wireless access point with a Raspberry Pi**

How to wirelessly connect to your Raspberry Pi

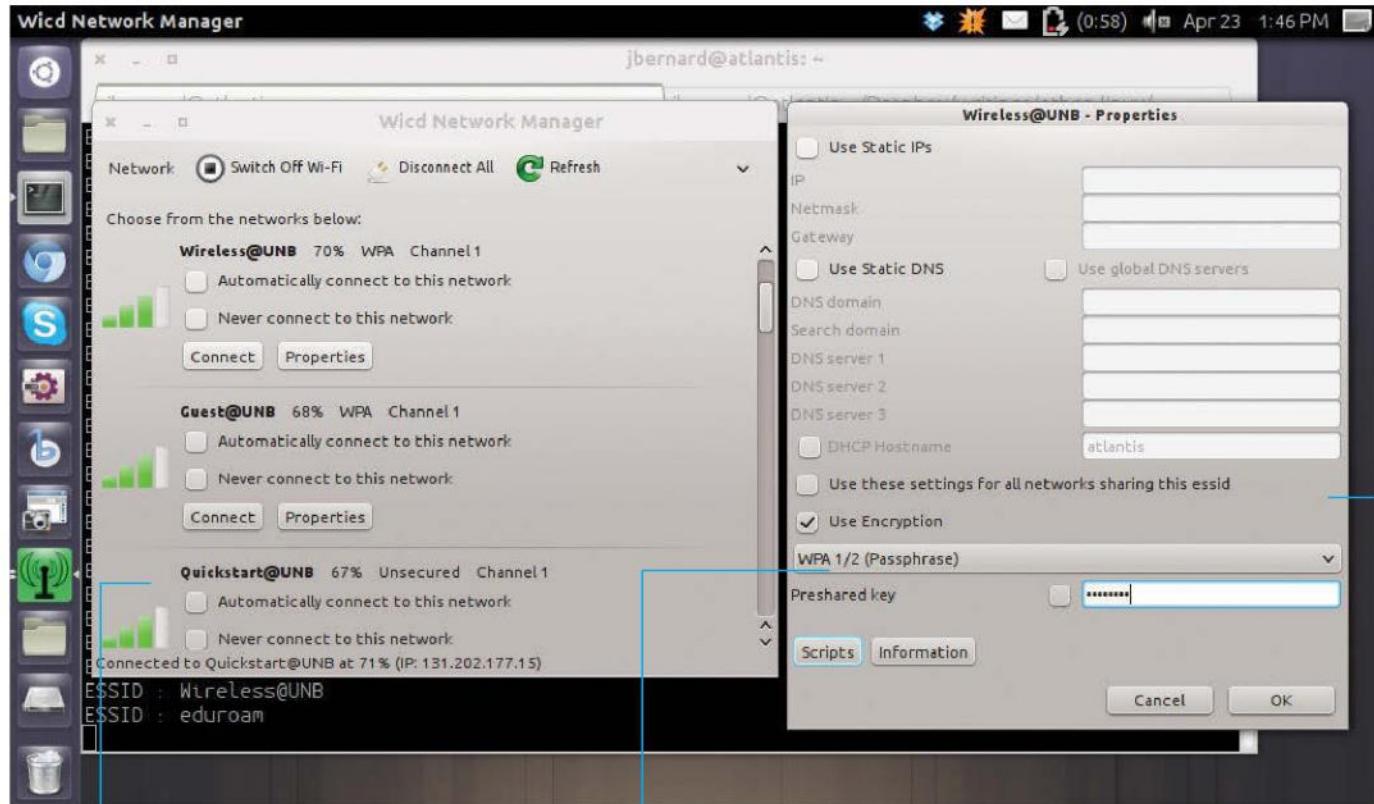


90

100

**“The Raspberry Pi can become an access point and have multiple devices connected to it”**





This is the main list of wireless networks available to you. You can select the network to connect to and set its properties.

Part of the properties you can set is the type of encryption being used, along with any special values that are needed, like passphrases.

When you click on the Properties button, you get a window where you can set static IP properties.

# Network wirelessly with wicd

Wicd is a flexible alternative to NetworkManager, complete with interfaces for GTK, KDE, curses and the command line. Use it to get your wireless network up and running

On most Linux systems, network management – both wired and wireless – is handled by a utility called **NetworkManager**. It is so ubiquitous that you may not even think about it. But, in Linux, there is always another choice. You can always do network management by manually configuring the appropriate configuration files. A better option is to use the utility wicd. Wicd provides interfaces using either GTK or KDE. This means you can use the

one appropriate for the graphic libraries for your desktop. There is also a text-based interface, which uses the curses library. You can even use wicd within your scripts or on the command line with the CLI interface. This tutorial will walk you through most of the interfaces, and how to use them to configure your machine's networking. This will include some issues, like using unusual setups of WPA security and adding functionality in the guise of network templates.

## Resources

**Wicd:** <https://launchpad.net/wicd>  
 or [wicd.sourceforge.net](http://wicd.sourceforge.net)



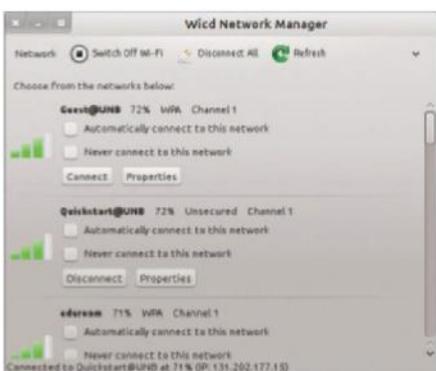
## 01 Get wicd

Wicd is hosted at both SourceForge and Launchpad – the URLs are provided on the previous page. On both websites you can find information on how to use wicd, as well as source code for the latest version.



## 02 Installation

Most distributions include a series of packages to install wicd. On Ubuntu each interface is available as a separate package. This means you can install only the portions that you need for your system. As always, you can install from source if you need the latest options.



## 03 The GTK interface

On most systems, you will likely want to use the GTK interface. To start it up, you can just type `wicd-gtk`. If your desktop has a tray, wicd will start up minimised to the tray. You can then click on it to open the main window. You can bypass the tray by using `wicd-gtk -n`.

## 04 Lists of wireless networks

When you open up wicd, it will try to pull up the wireless networks available to you. Depending on the specifics in your area, it may

miss some. You will want to click the refresh button to be sure that you pick up all of the networks available.



## 05 Preferences

There are general preferences that you can set in wicd. To get to them may not be obvious, depending on the default size of the main window. You may need to click on the arrow on the far right to display the other menu items available. On your advisor's system, this is where the Preferences option is located.



## 06 Interfaces

In wicd, you can only configure one interface at a time. You set this in the Interfaces section of the Preferences window in wicd. You can check to see which interfaces are available on your system with the commands `ifconfig` and `iwconfig`. Just running these with no options will give you those lists.



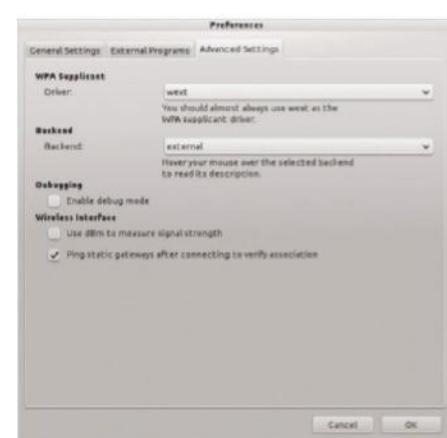
## 07 DNS servers

In the Preferences window, you can set global DNS options. This is useful if you want to use some other DNS server than that provided by your DHCP server. Or, if you are manually configuring the network details, you can set the DNS here.



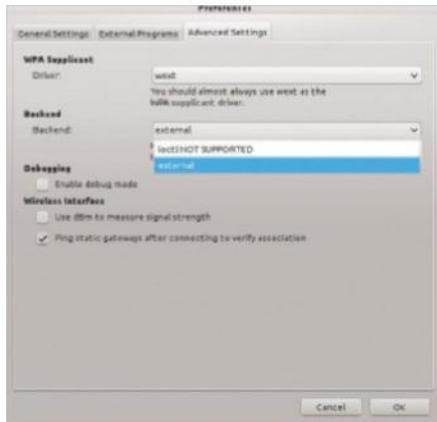
## 08 External programs

Selecting the External Programs tab of the Preferences window will allow you to set which external programs to use for various portions of the network configuration steps. It will query your system and only provide the options that are installed on your system. You can then select the specific programs for tasks like DHCP lookup.



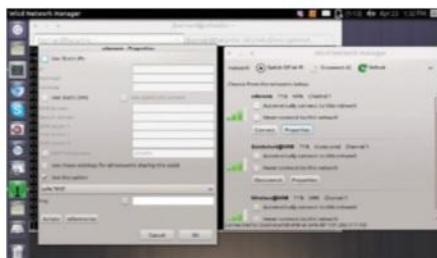
## 09 WPA supplicant program

WPA security is a bit of a bugbear. It is the preferred system to use, since WEP is so badly flawed. But, on Linux, it requires a separate program to handle the handshaking required. Clicking on the Advanced Settings tab in the Preferences window will allow you to select which program to use for WPA security.



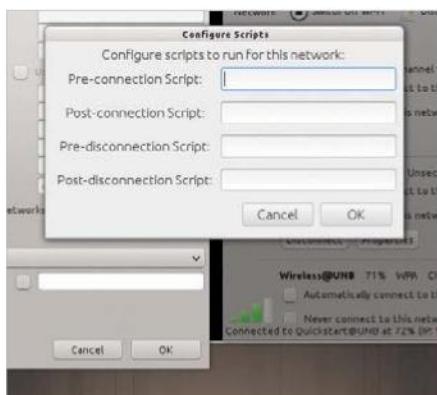
## 10 Back-end processing

Also in the Advanced Settings section is a selection for how to handle all of the back-end tasks to configure your network interfaces. The default (and most stable) is to use external programs, like iwconfig and dhclient. You can also choose to use IOCTL instead. It works faster, but is also more likely to fail.



## 11 Interface properties

Each available network has its own set of properties. You can pull up the Properties window by clicking on the Properties button. Here you can set options if you are using a static IP address. If you are using encryption, you can select from the list of possible templates at the bottom of the window.



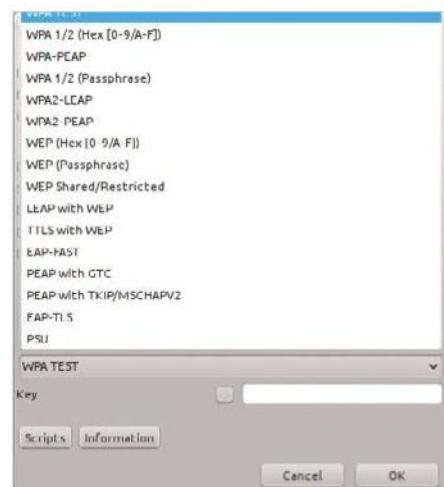
## 12 Scripts

At the bottom of the window, you also have the option of running scripts. There are options to set scripts to be run just before or just after connection, as well as just before or just after disconnection. This lets you customise connections to your needs.



## 13 Finding hidden networks

When you set up a Wi-Fi hotspot, you have the option of whether to broadcast the network name or to hide it from casual perusal. They'll still appear under wicd, labelled with the name <hidden>. This lets you find and connect to these hidden networks.

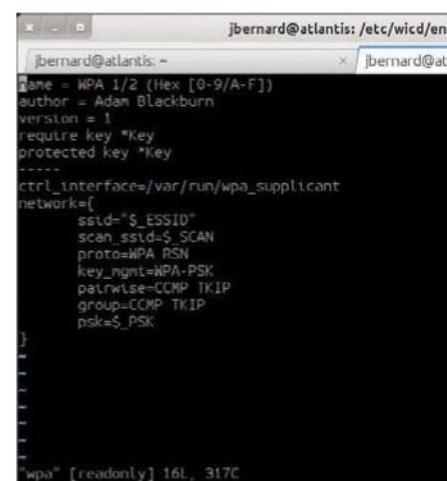
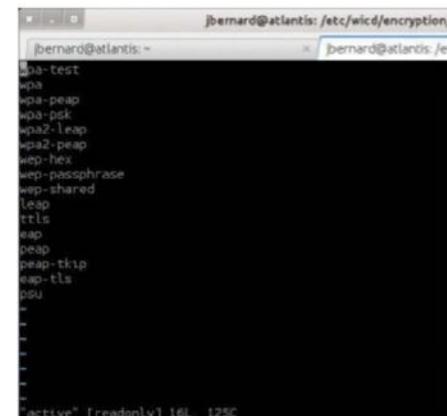


## 14 WPA templates

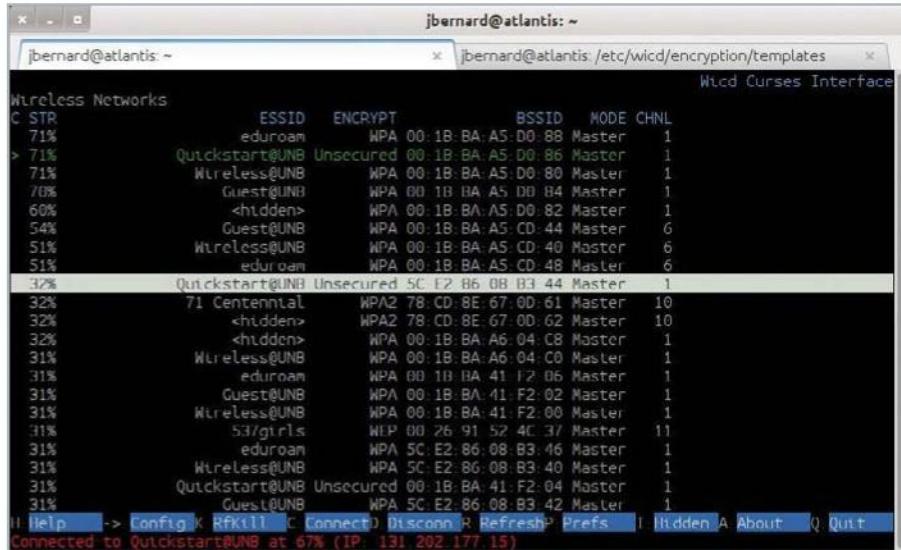
If the network you are using needs encryption, you can select the specifics from the list of available templates. These templates will change the remaining options in the Properties window and ask you for whatever values you need for that particular type of encryption.

## 15 Creating your own template

There are a surprising combination of options in encryption. So, wicd allows you to add templates for any combination of properties that wicd doesn't already support. The template files are stored in the directory /etc/wicd/encryption/templates. You should be able to find one that is already close to what you need. You can make a copy of this template and edit it to match the settings that you need. Once your new template is finished, you can add an extra entry in the file /etc/wicd/encryption/templates/active. It will then show up when you go to select the encryption template to use for your particular network.

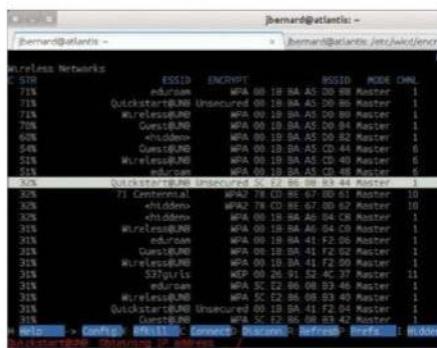


**“Setting up Wi-Fi at the console is where wicd really shines”**



## 16 Wicd-curses interface

There are several scenarios where you don't have a graphical interface but still need to set up wireless networking. Setting up WiFi at the console is where wicd really shines. There is a text interface using the curses library that gives you all the same functionality that is available in the GTK version.



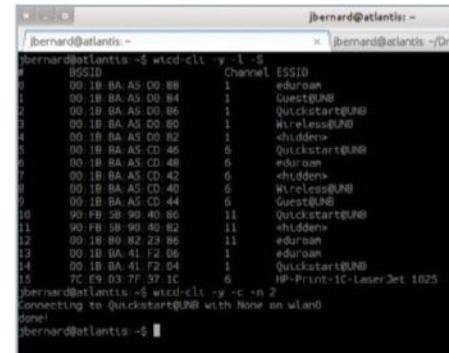
## 17 Connecting to a network

When the curses version starts up, it will show you the full list of available networks. You can use the arrow keys to move up and down the list to select the one you are interested in connecting to. When the correct one is selected, you can connect by pressing Enter.

## 18 Changing preferences

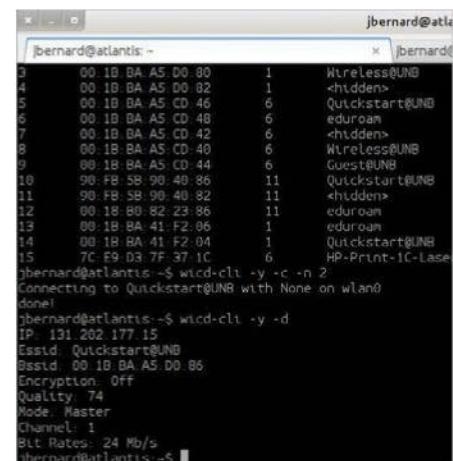
If the network you are interested in uses encryption, you need to set the template. To access the Properties window, you need to select the network of interest and then press the right-arrow key. You can then set any static elements, and also set the appropriate template.

anything. If you want to see the results, you can either add `-l` to the above command, or subsequently run `wicd-cli -y -l`.



## 21 Connecting to a network

To connect to a given network, you would use `wicd-cli -y -c -n NETWORKID`. Disconnecting is done equivalently with `wicd-cli -x -y`. Setting options is a bit more involved, where you need to set individual properties with `wicd-cli -y -p PROPERTY -s VALUE`.



## 22 Listing active connections

You can see the details of your current network connection by using the command `wicd-cli -y -d`. This includes the name, type of encryption, quality and bitrate, among other items.

## 23 Where to now?

Now, with wicd, you should have all the tools required to easily configure wireless networking, no matter how basic a console you have. You can even build scripts that can handle the connection details at the proverbial touch of a button.

Access Webmin via a web browser, locally or remotely

You can access Webmin from pretty much any platform. This tutorial covers installation methods that should work for most distros

The main window within the web browser is where you interact with most of the modules



# Manage your system with Webmin

As well as enabling you to administer a system remotely, Webmin is a great GUI front-end for system configuration

**Webmin offers administration of a Linux system via a web interface.** It is implemented as a set of Perl scripts that includes a small web server. Part of the appeal of Webmin is that it's extremely comprehensive as there are modules for most typical administration tasks. This includes core system areas such as management of printers and users, and package-specific tasks such as configuration of Apache and Squid. In addition, it includes some handy tools like basic file backup and transfer, and system resource monitoring.

Webmin isn't generally included in the package repositories for most Linux distributions, so setting it up entails downloading it from the website ([www.webmin.com](http://www.webmin.com)). This suits most

administrators since they tend to prefer carrying out updates of administrator tools when they are ready rather than as part of an automatic system upgrade. Webmin itself contains its own facilities for updating modules as updates become available.

So what can you do with Webmin? A typical example is configuring a system remotely, and we'll show you how to do that, but there's a lot more besides such mundane tasks on offer. Since it's so mind-bogglingly comprehensive, there is no reason why it can't be used as an all-encompassing system configuration GUI. In this context, it has the potential to offer a consistent organisation-wide configuration GUI that's backed up by a reassuring 15-year lineage.

## Resources

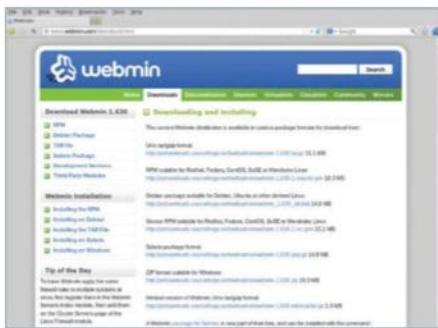
Any Linux system  
Webmin: [www.webmin.com](http://www.webmin.com)

## 01 Fetch the latest version

Head on over to [www.webmin.com](http://www.webmin.com) and proceed to the Download page to fetch the latest version of Webmin. Alternatively, download the latest version as a DEB file by typing

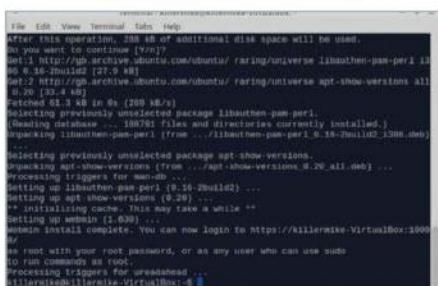
```
wget http://www.webmin.com/download/deb/webmin-current.deb.
```

Or type wget <http://www.webmin.com/download/rpm/webmin-current.rpm> for RPM.



## 02 Install Webmin (for DEB)

In the case of Debian-derived distributions, like Ubuntu, install the DEB file by typing `sudo dpkg -i [name of .deb you downloaded]`. This won't satisfy all of the required dependencies for the package, so don't worry if it throws up some errors. To fix this, type: `sudo apt-get -f install`



## 03 Install Webmin (for RPM)

In the case of Red Hat-derived distros such as CentOS, first make yourself root by typing su. To install, type: `yum install [name of RPM file you've downloaded]`.



The screenshot shows the Webmin login interface. On the left, there's a sidebar with links like 'Webmin', 'System', 'Servers', etc. The main area displays system information: System hostname (killermike-VirtualBox (127.0.1.1)), Operating system (Ubuntu Linux 13.04), Webmin version (1.630), Time on system (Wed Jul 10 01:06:50 2013), Kernel and CPU (Linux 3.8.0-25-generic on i686), Processor information (Pentium(R) Dual-Core CPU E5400 @ 2.70GHz, 2 cores), System uptime (0 hours, 48 minutes), Running processes (152), CPU load averages (0.03 [1 min], 0.15 [5 mins], 0.23 [15 mins]), CPU usage (0% user, 1% kernel, 0% IO, 99% idle), Real memory (496.04 MB total, 275.21 MB used), Virtual memory (510 MB total, 19.89 MB used), Local disk space (61.01 GB total, 5.40 GB used), and Package updates (All installed packages are up to date). Below this, a message says 'The following Webmin module updates are now available ...' followed by a table with columns 'Module', 'Version', and 'Fixes problem'. At the bottom, a note says 'Firefox automatically sends some data to Mozilla so that we can improve your experience.' and a 'Choose What I Share' link.

## 04 Connect to Webmin

Fingers crossed, Webmin is now working. You can test it by navigating to [https://\[your hostname\]:10000/](https://[your hostname]:10000/). You can discover your hostname by typing hostname at the command prompt. If everything is working, you should see the web interface for Webmin.

## 05 Log in with root

Typically, you will give Webmin your 'root' username and password, when prompted. On a system such as Ubuntu that has no root account, you can use the username and password of the regular user so long as that user can execute sudo.

The dialog box contains the text: 'You must enter a username and password to login to the Webmin server on 10.0.7.2.' It has fields for 'Username' and 'Password', and a checkbox for 'Remember login permanently?'. There are 'Login' and 'Clear' buttons at the bottom.

## 06 Set a static IP address

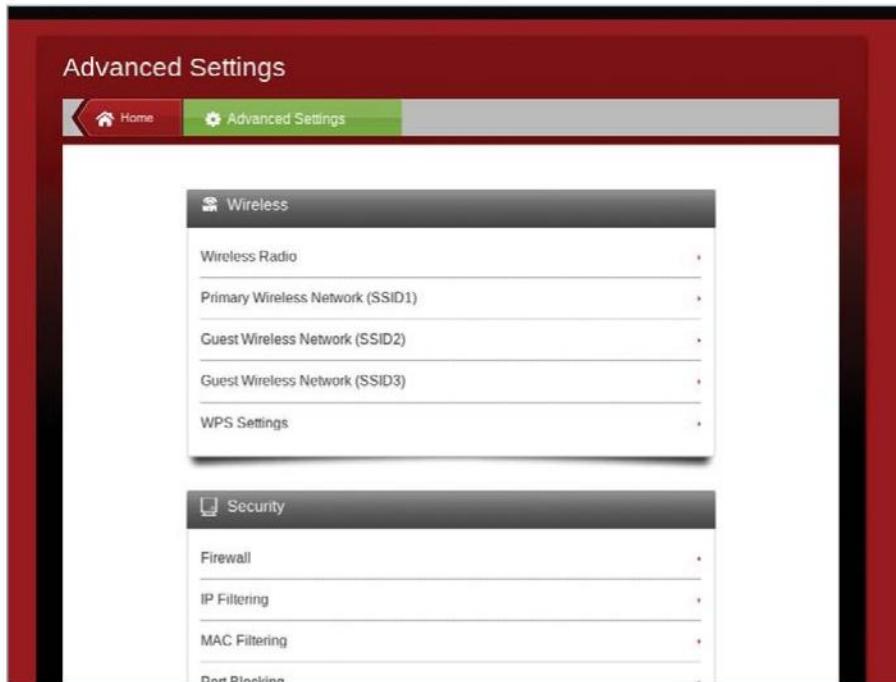
For convenience, you might want to assign a static IP address to the host machine, so that it can be consistently identified on the network. In Webmin, go to Networking>Network Configuration>Network Interfaces to see a list of currently activated network interfaces. Click on the name of the network adaptor used to connect to the rest of your network ('eth0' or

similar for a wired network; 'wlan0' for a WiFi one). From here, change the IP address to something congruent to your current numbering scheme but higher. For example from 192.168.1.5 to 192.168.1.200.

This screenshot is identical to the one above, showing the Webmin dashboard with system information and module update details. The main difference is that the connection status is now successful, allowing access to the full range of management features.

## 07 Configure your local firewall

If you can't access Webmin from other machines on your network (by using [http://\[IP address\]:10000/](http://[IP address]:10000/)), don't panic: it probably means a local firewall is blocking port 10000. You can configure the local firewall using Webmin itself, too. Go to Networking> Local Firewall and click on the add button. Most of the fields on the next page can be ignored, but set Network Protocol to equals TCP and Source TCP port to equals 10000. Click on Create to apply.

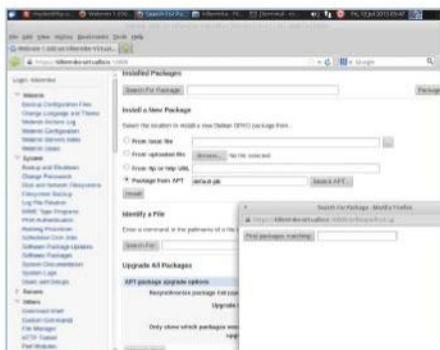


## 08 Configure your router

How you enable access from the internet is specific for each router, so you'll need to examine the instructions for your router or visit [portforward.com](http://portforward.com). In short, you need to allow incoming transmissions for port 10000 and forward that port to the IP address of the host computer on your network. This is usually very straightforward, but since every router is different, it's impossible to explain in detail here.

## 09 Webmin package management

In this example, we'll install OpenJDK to a stock Ubuntu machine using the Webmin package management facilities. Open the System submenu from the sidebar and select Software Packages. Note that the Search... button opens a window that allows searching of the APT database.

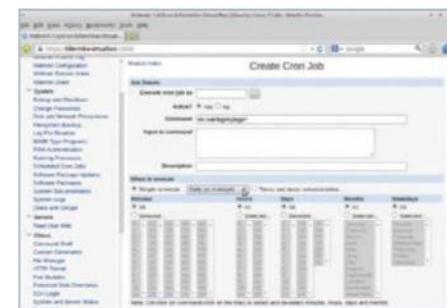


## 10 Package Installation

We'll install Java support on Ubuntu in this example, but things work in much the same way on other platforms. On the main page, select the package from APT and enter default-jdk into the text box. Now select Install. Like many of the Webmin modules, this window contains quite a lot of useful text output. At the top it shows the Apt command that it shall execute (apt-get -y --force-yes -f install default-jdk in this case). Beneath this, it shows the output of Apt and below this, a tabular summary of all packages installed.

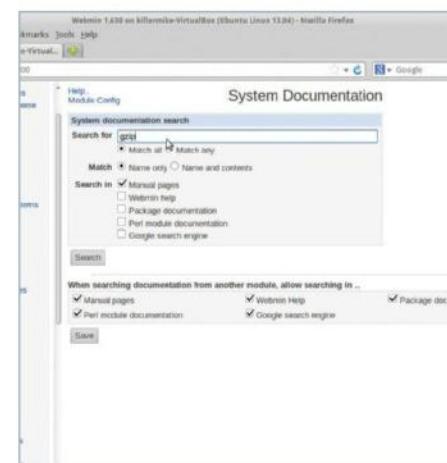
## 12 Create a cron job

We're going to create a nightly job that deletes the contents of /var/log/mylogs using the Scheduled Cron Jobs page. Select the 'Create a new scheduled cron job' option and this takes you to the cron job editing page. In the job details section, specify that the job will be executed by user root. Type rm /var/log/mylogs/\* into the Command text box. Add a description for the job to the Description box. In the 'When to execute' section, select Simple schedule and Daily (at midnight). Click on Create.



## 13 Search system documentation

The System Documentation module (accessed from within the System section), offers a complete man page browser, usefully, accessing the man files on the host system. To use it, simply enter a search term. It operates much the same way as the standard 'man' command, but many will find the web-based interface more convenient.



## 11 Manage cron jobs

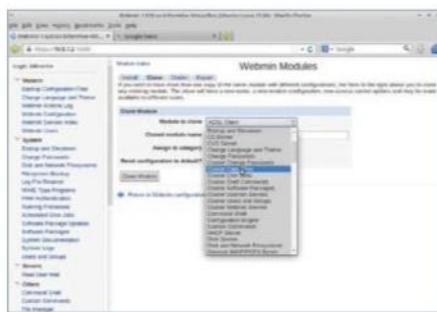
The syntax of specifying a new scheduled UNIX job, or cron job, is notoriously fiddly, and Webmin offers a neat front-end for this. Begin by entering the System>Scheduled Cron Jobs page. On this page, you can create new jobs or edit jobs already on the system.

## 14 View logs

The Webmin Actions Log is a feather in Webmin's cap. If you don't add anything to the default form and click on search, you will be given a summary of configuration changes made that day. You can further refine your search by, for example, widening the date range.

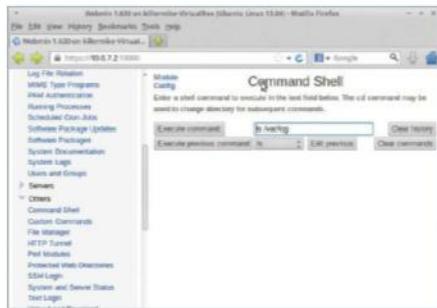
## 15 Clone module

Clone a module to have a second (or more) version with different settings. Go to Webmin > Webmin Configuration > Webmin Modules. Now select the Clone tab. From here, select a module to clone from the drop-down and give it a new name. Click on 'Clone module' and it will be added to the sidebar.



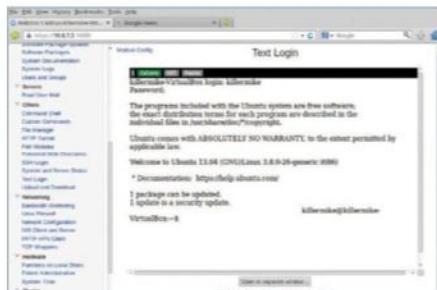
## 16 Execute commands

Webmin has basic facilities for the execution of commands on the host machine, with a display of the output using the Command Shell page. If there's a command that you want to assign to a clickable button for frequent use, add it in the Custom Commands page.



## 17 Text login

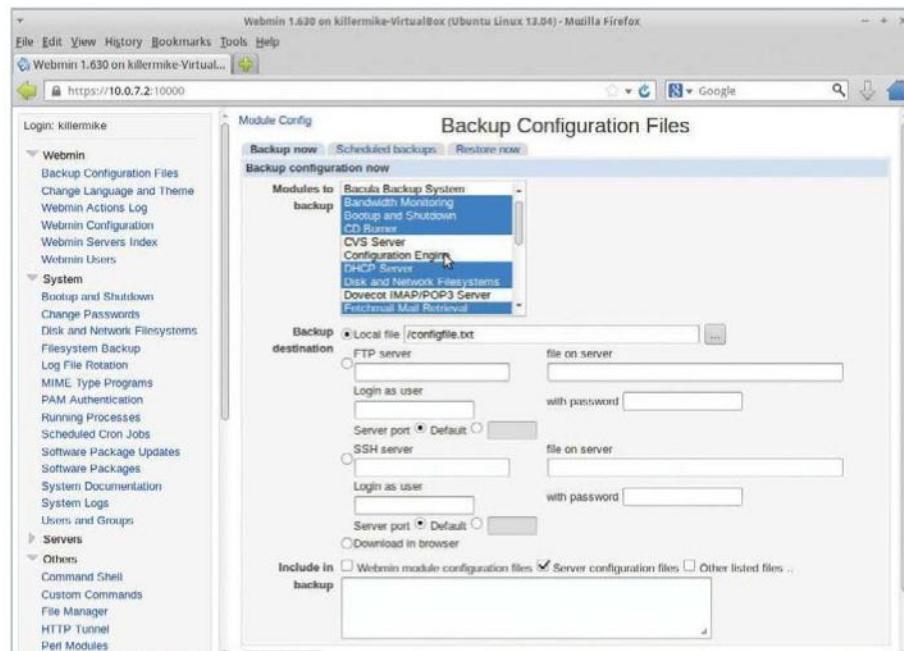
If you need a bit more flexibility than the execute commands page allows, but you don't want to (or aren't able to) install SSH, try Others > Text Login. This offers a full terminal in which you can run text-mode programs, all from within the browser.



# "The Webmin Actions Log is a serious feather in Webmin's cap"

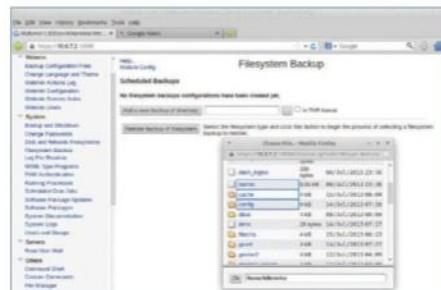
## 18 Back up config files

You can back up any configuration that Webmin modules can access to a local or remote file for later restoration on the Webmin > Backup Configuration Files page. You can independently select which modules to back up and restore (Ctrl- or Shift-click), and there's a facility for scheduled backups.



## 19 File system backup

Webmin includes a simple facility to back up files and directories. To use it, go to System > Filesystem Backup. Specify a directory and click on 'Add backup directory'. From here you can specify details such as a schedule for the backups and the remote (SSH, FTP) or local destination for the archive.



## 20 Monitor bandwidth

Go to Networking > Bandwidth Monitoring and click on 'Set up now'. Once it's set up, you have to create some traffic on that interface before a report can be generated. When you've done this, specify a time range and click on 'Generate report'. Note that this function has a performance overhead.

## 21 Add a disk-space monitor

Webmin monitors alert you when certain conditions are met. Go to Others > System and Server Status. Click on Scheduled Monitoring and specify your SMTP email details. Select Disk Space from the drop-down and specify a 5% minimum size. Click on Create. Congratulations, you've started to master Webmin!



Unison synchronises files between computers – here we're using the command-line version of the tool

This line indicates that Unison has nothing to sync at the moment, since no files have changed

This error message shows that there is a lock file that needs to be manually removed

```

Terminal Shell Edit View Window Help
mtsouk@li140-253: ~ code — bash — 75x37
Connected //MacBookPro.local//Users/mtsouk -> //li140-253//home/mtsouk]
Looking for changes
Waiting for changes from server
Reconciling changes
local      li140-253
changed --> docs/article/working/unison.LUD/article.doc [f]
new file -->   docs/article/working/unison.LUD/figures/PermitUser.r.png [f]
Proceed with propagating updates? [y]
Propagating updates

UNISON 2.40.102 started propagating changes at 16:35:23.47 on 02 Aug 2013
[END] Updating file docs/article/working/unison.LUD/article.doc from /Users
/mtsouk to //li140-253//home/mtsouk
[END] Copying docs/article/working/unison.LUD/figures/PermitUser.png from /
Users/mtsouk to //li140-253//home/mtsouk
[END] Updating file docs/article/working/unison.LUD/article.doc
[END] Copying docs/article/working/unison.LUD/figures/PermitUser.png
UNISON 2.40.102 finished propagating changes at 16:35:30.40 on 02 Aug 2013

Saving synchronizer state
Synchronization complete at 16:35:31 (2 items transferred, 0 skipped, 0 fo
iled)
MacBookPro:code mtsouk$ unison articles
Contacting server...
Connected //MacBookPro.local//Users/mtsouk -> //li140-253//home/mtsouk]
Looking for changes
Waiting for changes from server
Reconciling changes
Nothing to do: replicas have not changed since last sync.
MacBookPro:code mtsouk$ 

MacBookPro:~ mtsouk$ unison docs
Contacting server...
Looking for changes
Fatal error: Warning: the archives are locked.
If no other instance of unison is running, the locks should be removed.
The file /Users/mtsouk/.unison/lkefb86c6e19a1a28782cd0633cc173eb7 on host MacBoo
KPro.local should be deleted
The file /Users/mtsouk/.unison/lk58fd5a518a7395268005323a8cabcf71d on host MacBoo
KPro.local should be deleted
Please delete lock files as appropriate and try again.
MacBookPro:~ mtsouk$ 

MacBookPro:~ mtsouk$ unison articles
Contacting server...
ssh: connect to host linode port 22: No route to host
Fatal error: Lost connection with the server
MacBookPro:~ mtsouk$ 

```

# Synchronise your files with Unison

Learn how to use the Unison command-line tool to synchronise files between computers quickly and reliably

## Resources

**Unison:** [www.cis.upenn.edu/~bcpierce/unison](http://www.cis.upenn.edu/~bcpierce/unison)  
**Two networked UNIX machines**

**Unison** is an open source file synchronisation tool for both text and binary files. It also has a GUI, but here we're focusing on only the command-line version because it's quicker and gets the job done cleanly. Unison really shows its capabilities when you are working with more than one computer and you need synchronisation across all of them.

Benjamin C Pierce led the creation of Unison at the University of Pennsylvania and it started life as a research project. It can be used through the

SSH service and works equally well on both UNIX (Linux, Mac OS X, etc) and Windows machines.

It should be apparent that Unison was inspired by the rsync utility. Unison differs from rsync in that the latter is a mirroring tool that needs to know in advance where the willing-to-keep versions of the files are, whereas Unison is a synchronisation tool that identifies the files that have been changed since the last sync process and decides the way that the changes are going to be propagated. In short, it's smart.

```

root@li140-253:~# apt-cache search unison
unison-all - file synchronization tool (all console versions)
unison-all-gtk - file synchronization tool (all GTK+ versions)
mr - Multiple Repository management tool
unison - file-synchronization tool for Unix and Windows
unison-gtk - file-synchronization tool for Unix and Windows with GTK+ interface
unison2.27.57 - file-synchronization tool for Unix and Windows
unison2.27.57-gtk - file-synchronization tool for Unix and Windows with GTK+ interface
unison2.32.52 - file-synchronization tool for Unix and Windows
unison2.32.52-gtk - file-synchronization tool for Unix and Windows with GTK+ interface
root@li140-253:~# apt-get install unison
Reading package lists... Done
Building dependency tree
Reading state information... Done
unison is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@li140-253:~

```

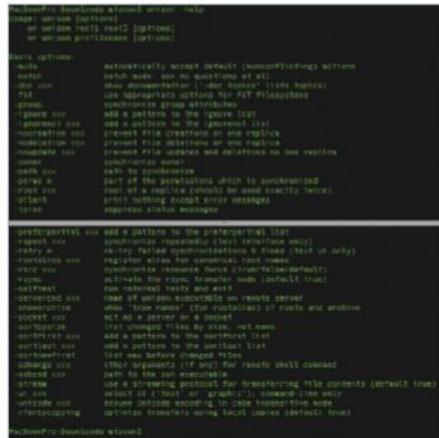
## 01 Installing Unison

Most Linux distributions have Unison as a package ready for installation so that you do not have to compile it. At the time of writing, the current stable version of Unison is 2.40.102. Note that every machine that is part of the synchronisation process must have a copy of the command-line version of Unison installed. Additionally, this copy of Unison should be located inside one of the directories of the default PATH shell variable. Assuming you're on a Debian-based system, simply type:

```
sudo apt-get install unison
```

To find the version of Unison you are using, just type the following command:

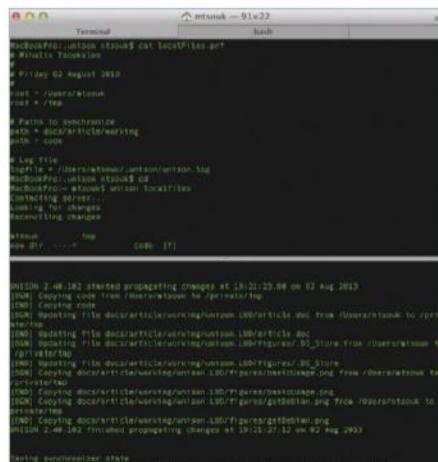
```
unison -version
```



## 02 Unison profile files

Unison can run from the command line without using any configuration files (profiles), but using a profile greatly simplifies its use –

having a profile file does not prevent you from adding extra command-line options (although this article will not deal much with such options). In the rare event that you have troubles working with Unison, you may run it using the ‘-debug all’ command-line option so that you can better trace and resolve errors.



## 03 Synchronising files on the same computer

Although Unison was initially developed with synchronising files located on different machines in mind, you can synchronise files located on the same computer using one of the following two methods:

- Using localhost as the remote machine name.
- Using only local directory paths for both root = lines.

Your advisor prefers the second way as it does not require the SSH server be up and running.

The Unison profile file is called localFiles.prf

and is located inside the ~/.unison directory. The following lines to work on the same machine, as they do not include a remote machine:

```
root = /Users/mtsouk
root = /tmp
```

## 04 Synchronising files with another computer

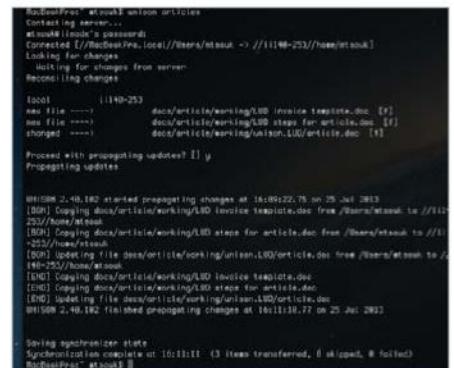
Given a profile name called articles.prf that is located inside the ~/.unison directory, you can tell Unison to use it by executing the following command:

```
$ unison articles
```

The remote machine declaration starts with

```
root = ssh://.
```

There are rare occasions – usually when the user changed a file on both computers before synchronising – where Unison will not be able to determine whether a file or directory has changed on the local or the remote server. In such situations, Unison kindly asks for your help so that it will not mistakenly proceed using the wrong version.



```

articles.prf — .unison
articles.prf

1 # Mihalis Tsoukalos
2 #
3 # Friday 05 July 2013
4 #
5 root = /Users/mtsouk
6 root = ssh://linode//home/mtsouk
7
8 # Paths to synchronize
9 path = docs/article/working
10
11 # Log file
12 logfile = /Users/mtsouk/.unison/unison.log
13
14 # Backup files
15 backup = Name *
16 ignore = Name *.DS_Store
17 ignore = Name .DS_Store
18
19

```

## 05 Running Unison using a simple profile file

Some basic things you need to know about Unison profiles are:

- The default backup level is 2, which controls how many previous versions of each file are kept not including the current version of the file. So the default backup level keeps two backup copies of a file. The parameter for setting the backup level is called maxbackups.
- You can tell Unison not to keep file backups by not adding any backup options inside a Unison profile file.
- Profile files support Unicode characters.
- The logfile option tells Unison to keep a log file called unison.log inside the .unison directory.
- Sometimes, network problems prohibit file synchronisation, but Unison's error messages are denoting the problem:

```
$ unison articles
Contacting server...
Connection closed by 109.xxx.yyy.zzz
Fatal error: Lost connection with the server
```

computer) that are going to participate in the syncing process as well as the directories that are considered the root directories for each machine. After those declarations, the actual directories that are going to be synced are listed. In this example we have just one directory (one path = line). Its full path is /Users/mtsouk/docs/article/working for the local machine and /home/mtsouk/docs/article/working for the machine called linode (an IP alias inside /etc/hosts). You can have as many path entries as you want. All files are synced except the ones that match the ignore option. You can also have as many ignore = lines as you want and can use regular expressions in them. The backup option tells Unison to keep backups of all files.

```

mtnouk@mtnouk: ~ $ unison articles
starting server...
contacting server...
waiting for changes
consulting changes
  11140-253
  * dir: docs/article/working/unison.LDB [article.uno.0f]
  * file: docs/article/working/unison.LDB/figures/unison.png [f]
need with propagating updates? [] y
propagating updates

UNISON 2.40.102 started propagating changes at 19:09:04.19 on 31 Jul 2013
UNISON 2.40.102 updating file docs/article/working/unison.LDB/article.uno from /Users/mtsouk to //10.0.0.100/home/mtsouk
UNISON 2.40.102 copying docs/article/working/unison.LDB/figures/unison.png from /Users/mtsouk to //10.0.0.100/home/mtsouk
UNISON 2.40.102 copying docs/article/working/unison.LDB/figures/unison.png
UNISON 2.40.102 skipping file docs/article/working/unison.LDB/article.uno
UNISON 2.40.102 finished propagating changes at 19:09:07.21 on 31 Jul 2013

syncing synchronizer state
synchronization complete at 19:09:08 (2 items transferred, 0 skipped, 0 failed)
UNISON 2.40.102 unison articles
UNISON 2.40.102 contacting server...
  11140-253
  * dir: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
  * file: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
  * file: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
UNISON 2.40.102 waiting for changes
UNISON 2.40.102 contacting server...
UNISON 2.40.102 syncing changes
UNISON 2.40.102 nothing to do: replicas have not changed since last sync.
should run: mtsouk#
```

## 06 Explaining the profile file

Lines starting with a # denote comments and are not processed any further. Exactly two lines must start with root =. They declare the machines (one is always the local

## 07 The .unison directory

Most of the Unison housekeeping happens inside the .unison directory which resides in the user's home directory.

Unison keeps the following data there:

- Its profile files including the default profile called default.prf.
- Its backup files, if you decided to have central backups.
- The status file for each used profile.
- The Unison log file, should you tell it to keep one. You can put it anywhere you want but it is very convenient for it to reside there.
- Unison creates lock files there, during synchronisations, which it deletes when finished. **Note:** It is not recommended to synchronise the whole .unison directory. Synchronising just the profiles is okay.

## 08 A more advanced Unison profile file

The contents of the advanced.prf file are as follows:

```

root = /Users/mtsouk
root = ssh://linode//home/mtsouk
times = true
batch = true
# Log file
logfile = /Users/mtsouk/.unison/unison.log
# Paths to synchronize
path = code/C
# Backup files
backup = Name *
backuplocation = central
maxbackups = 3
ignore = Name {Thumbs.db}
ignore = Name {*.DS_Store}
ignore = Path {Samples.lnk}
```

```

mtnouk@mtnouk: ~ $ unison advanced -testserver
internal error: File "advanced", line 5: bad pattern:
pattern must be introduced by one of the following keywords:
  Name, Path, BelowPath or Regex.
mtnouk@mtnouk: ~ $ unison advanced -testserver
contacting server...
  11140-253
  * dir: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
  * file: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
  * file: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
  * file: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
UNISON 2.40.102 started propagating changes at 23:03:18.00 on 02 Aug 2013
UNISON 2.40.102 copying code/C from /Users/mtsouk to //11140-253//home/mtsouk
  11140-253
  * dir: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
  * file: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
  * file: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
  * file: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
UNISON 2.40.102 finished propagating changes at 23:03:21.21 on 02 Aug 2013

syncing synchronizer state
synchronization complete at 23:03:21.21 (1 item transferred, 0 skipped, 0 failed)
UNISON 2.40.102 unison advanced
UNISON 2.40.102 contacting server...
  11140-253
  * dir: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
  * file: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
  * file: /MacBookPro.local//users/mtsouk --> //11140-253//home/mtsouk
UNISON 2.40.102 waiting for changes
UNISON 2.40.102 contacting server...
UNISON 2.40.102 syncing changes
UNISON 2.40.102 nothing to do: replicas have not changed since last sync.
should run: mtsouk#
```

```
ls -l
ls -l .ssh/id_rsa.pub
cat advanced.prf
```

```
# Michael Tsoukatos
# Friday 05 July 2013
#
# root = /Users/mtsouk
root = ssh://linode/home/mtsouk

times = true
batch = true

# Log file
logfile = /Users/mtsouk/.unison/unison.log

# Paths to synchronize
path = code/C

# Backup Files
backupdir = .
backuplocation = central
maxbackups = 3
ignore = Name {'.DS_Store'}
ignore = Name {'.DS_Store'}
```

## 09 Explaining the advanced profile file

- The `times = true` line tells Unison to synchronise modification times.
- The `maxbackups = 3` line tells Unison to keep the current file version plus three backups of it.
- The `backup = Name *` line tells Unison to back up every file.
- The `backuplocation = central`, which is the default option, tells Unison to keep all backups in a central location. If neither the `backupdir` preference nor the environment variable `UNISONBACKUPDIR` are set, the `.unison/backup` directory is used as the backup location. If set to local, then all backups will be kept in the same directory as the original files.
- The `batch = true` option is a little tricky and you should be careful with it as Unison will ask no questions at all and non-conflicting changes will be propagated whereas conflicts will be skipped. Nevertheless, it is an essential option if you want to use Unison as a cron job.
- The `ignore = Name {'.DS_Store'}` line tells Unison to not synchronise files that end with `.DS_Store`.

## 10 Using SSH without giving a password

The single most time-saving thing you can do is to set up SSH so that you will not need to enter your password each time you want to synchronise your files and directories. The procedure is easy and involves the following three steps:

### 1. Run `ssh-keygen -t rsa`

You will have to enter a passphrase twice, so please do remember the passphrase! Two new files are going to be created: `~/.ssh/id_rsa` and `~/.ssh/id_rsa.pub`.

### 2. You may need to create a directory called `.ssh` on the remote server if it does not already exist.

3. Copy the contents of the `~/.ssh/id_rsa.pub` file from your local server into the file `~/.ssh/authorized_keys` found on the remote server. One way of doing it is by executing the following command:

```
$ cat ~/.ssh/id_rsa.pub | ssh linode 'cat >> .ssh/authorized_keys'
```

The next time you try to log into the remote Linux server using SSH, you will be asked for the passphrase of step 1 for the last time.

From now on, you can log into the remote Linux server by just typing `ssh linode`:

```
$ ssh linode
Linux (none) 3.9.3-x86_64-linode33
#1 SMP Mon May 20 10:22:57 EDT 2013
x86_64
.
.
.
Last login: Wed Jul 31 18:46:23 2013
from ppp-94-64-21-97.home.otenet.gr
mtsouk@li140-253:~$
```

The first time you log into the remote server without typing your password, the following informative message will be on the screen:

```
Identity added: /Users/mtsouk/.ssh/
id_rsa (/Users/mtsouk/.ssh/id_rsa)
```



## 11 Two common Unison troubleshooting techniques

There are times when things do not work as expected. Unison offers you many options that can help you both find and solve problems.

The first option to try is the `-testserver` option that just connects to the remote server and then exits without synchronising any files.

The second thing to do is run the following command:

```
$ ssh remote.machine.domain 'echo $PATH'
```

The aforementioned command let you see whether the PATH is the same as when you log in using `ssh remote.machine.domain`. If the problem is with the PATH, check if the option `PermitUserEnvironment` in `/etc/ssh/sshd_config` is set to 'no' and change it to 'yes'.

```
which unison
pt/Local/bin/unison
checkProc: mtsouk ssh localhost 'echo $PATH'
ssh: command not found
telnet error: Lost connection with the server
checkProc: mtsouk
```

## 12 Unison hints and tips

The first two or three times you use a new profile, double-check if everything works as expected.

- You do not need to use every parameter that Unison supports, just the ones that will do your job!
- You can troubleshoot Unison using the `-debug all` command-line option. It will generate lots of output useful for debugging.
- The more you use Unison, the more you will understand its practicality.
- You should be very careful with your backup options, especially maxbackups, as it can take up too much space on your computer.
- You can use Unison to securely exchange files between computers.
- If a Windows machine is involved in the synchronisation process, be careful with file and directory permissions.
- For non-critical data files you may run Unison once a day, but for critical data you should run it more often.
- Unison cannot replace regular backups!
- When you are making a new profile, either start simple or use an existing one as a starting template. Add the extra functionality and features while making sure that you always have a working profile.

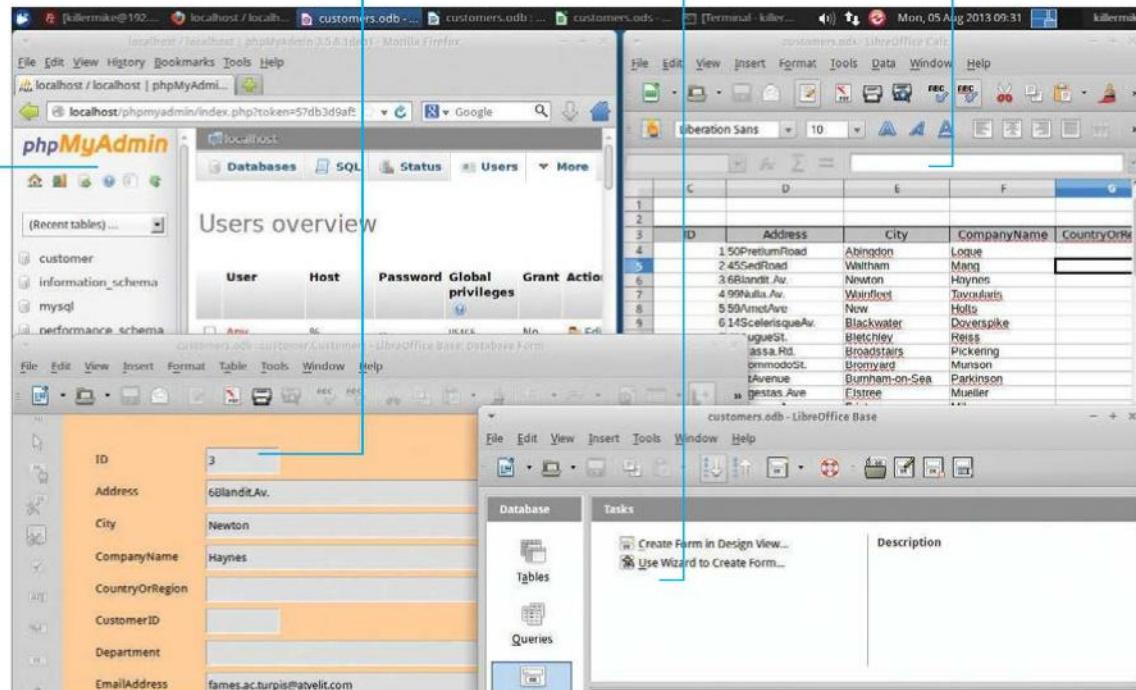
```
unison@li140-253:~$ ssh remote.machine.domain 'echo $PATH'
MacBook-Pro:~ mtsouk$ unison articles -debug all
[startup] Preferences:
allowGraphic = false
host = localhost
server = false
prefdock = false
dock = false
version = false
silent = false
dubtify = false
testserver = false
rest = articles
showprey = false
selftest = false
configure = false
empty = 0
repeat = 0
contactquietly = false
key =
label =
.
.
.
remote = false
rsrc = default
msecver = 1
unicodedS = false
unicodedU = false
unicode = default
msecdot11inessitive = false
ignorecase = default
timers = false
terse = false
logFile = /Users/mtsouk/.unison/unison.log
log = true
debugLines = false
debug = all
addressInfo =
Contacting server...
ssh: [Errno 2] No such file or directory
ssh: connect to host linode port 22: No route to host
Fatal error: Lost connection with the server
MacBook-Pro:~ mtsouk$
```

phpMyAdmin offers a web-based front-end for the creation and maintenance of MySQL databases

Data entry is carried out via an easy-to-use form in Base, the front end to our database

All of the actual database design (fields, form layout etc) is carried out from within Base

You import and export data to and from LibreOffice Base by using Calc, the spreadsheet module. This enables access to most common data formats



# Make a small business database with LibreOffice

Create a database that combines an easy-to-use, form-based front-end using LibreOffice with a portable, networked MySQL back-end

We're going to show you how to put together a typical database for small business use: a database of customer details. It will be possible to both export and import contact data in standard formats by making use of Calc, LibreOffice's spreadsheet module. We'll use Gmail contacts as our source, but you can use any software that can export CSV files – and pretty much everything can.

We've added a few twists to keep things interesting. This project uses the Base module of LibreOffice as the front-end, and this provides a GUI for setting up the database, creating the

forms for data entry and the actual business of entering data. For the back-end, we will be using the industry-standard MySQL. This allows us to locate the back-end on a central server. This, in turn, allows multiple users to access the database.

For initial creation of the MySQL database, we'll use phpMyAdmin thanks to its friendly web interface, although the actual database design will be carried out from within Base. By the end of the project, you will have a GUI system for browsing and editing the database with a portable, networked back-end.

## Resources

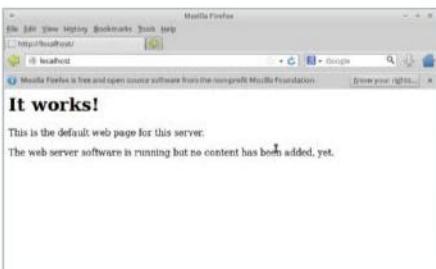
LibreOffice: [www.libreoffice.org](http://www.libreoffice.org)

At least one Ubuntu Linux PC



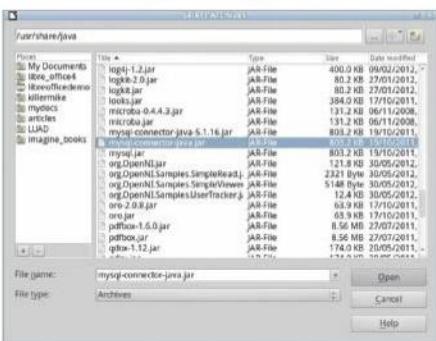
## 01 Install LibreOffice

At time of writing, the major Linux distributions haven't moved over to LibreOffice 4 and are still offering 3.x. This means that you may have to install LibreOffice 4 manually. Visit the LibreOffice website ([www.libreoffice.org](http://www.libreoffice.org)) and follow the instructions. On Ubuntu, this consists of unpacking the archive and running `sudo dpkg -i *.deb` on the contents.



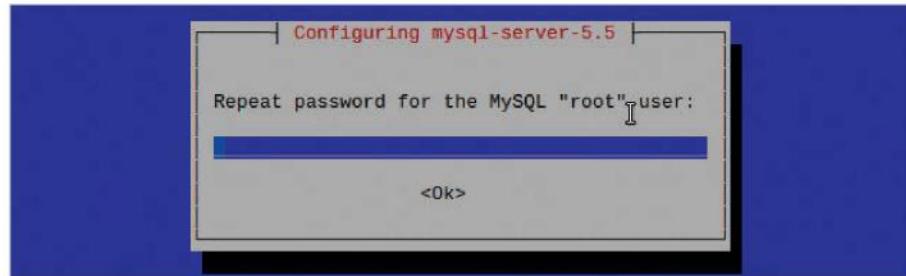
## 02 Install the Apache web server

We'll install Apache early on and with its own command because some of the other packages need to be able to configure a working installation. Carry out the installation with `sudo apt-get install apache2`. Test it by navigating to `http://localhost`.



## 03 Install Java and additional classes

Connectivity between Base and MySQL makes use of a Java class. Type `sudo apt-get install default-jdk` to install the Java runtime. Type `sudo apt-get install libmysql-java` to install the needed additional Java classes.



## 04 Install the MySQL Server

Type `sudo apt-get install mysql-server` to begin installation. Before long, you should be prompted to set a root MySQL password. Note this isn't the same as the administrator account of your system, which is also called 'root'. Choose a password and make a note of it.

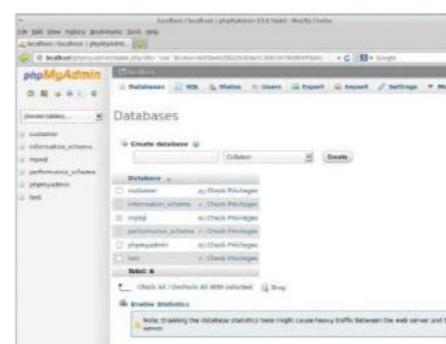
## 05 Install and test phpMyAdmin

Type `sudo apt-get install phpmyadmin` to begin installation. When prompted to choose a web server, choose Apache2, select it with the space bar and press Return. When requested, give it the MySQL root password and then choose a password for phpMyAdmin and make a note of it. Navigating to `http://localhost/phpmyadmin/` should take you to a functioning login page. Log in using the MySQL root username and password. We'll use MySQL to set up and maintain the actual database, although we'll create the fields from within LibreOffice later on.



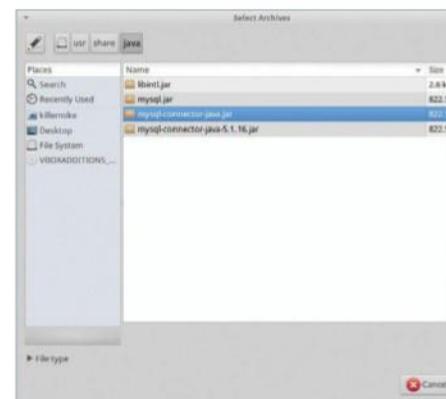
## 06 Create database

Within the phpMyAdmin web interface, select the Databases tab. Now create a new database by entering the name 'customer' into the text box and clicking on Create. This database will contain our customer data.



## 07 Add JDBC in LibreOffice

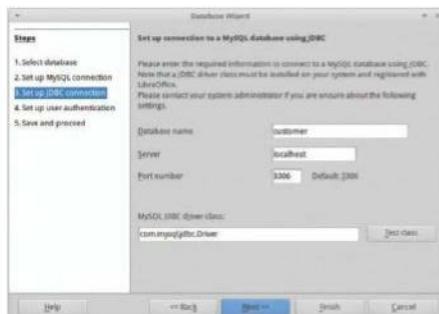
We now need to tell LibreOffice where to find the JDBC class file. Start LibreOffice and go to LibreOffice>Options>Advanced. In the Java Options section, select Class Path and then Add Archive. The file you need is located at `/usr/share/java/mysql-connector-java.jar`. Select it and restart LibreOffice.





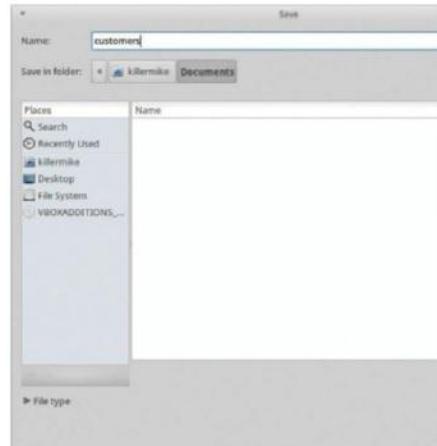
## 08 Connect the database

We now need to connect our front-end (LibreOffice) to the back-end (MySQL) of our database. Start LibreOffice and launch the Base module. In the dialog that pops up, select 'Connect to an existing database'. From the drop-down menu below this, select MySQL as the database type.



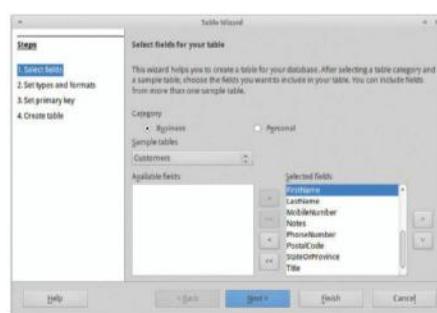
## 09 Configure Base

On the next page, select 'Connect using JDBC'. On the next page, click on 'Test class' to ensure that the Java RT is working. Now enter the name of the database that we created, customer, and enter localhost into the Server field. On the next page, add Root as the username and tick 'Password required'. Now click on the 'Test connection' button and enter the root MySQL password, when asked for it, to test the connection between LibreOffice and the local MySQL server. Presuming that this completes without errors, click on Next.



## 10 Save the database

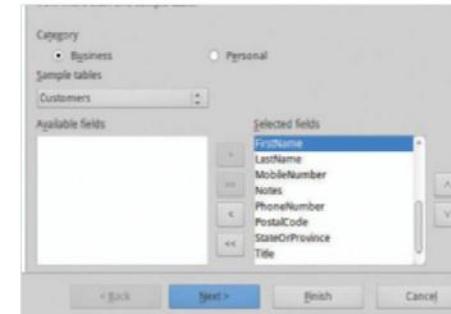
You can accept the defaults on the next page, so click on Finish. When prompted, give the database a name and save it. Remember that this file contains the connection information for access to our MySQL database – it doesn't contain the actual records.



## 11 Create fields from a template

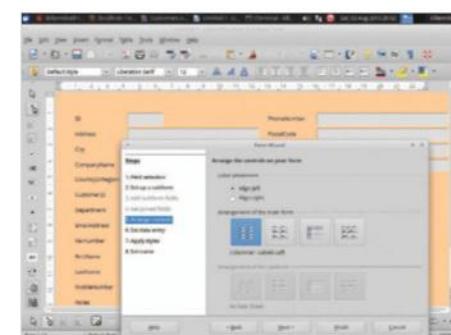
You may want to create a custom set of fields for your customer records, but to save time we're going to use the one of the templates that is built into Base. Select Tables from the side menu and then 'Use Wizard to Create Table'. Using the Sample tables pull-down menu, select Customers. Use the >> button to copy all of them across. On the next page, you can tweak the fields that you have included and add new ones. Select the defaults on the next two pages and then click on Finish.

**“Remember that this file contains the connection information for access to our MySQL database – it doesn’t contain the actual records”**



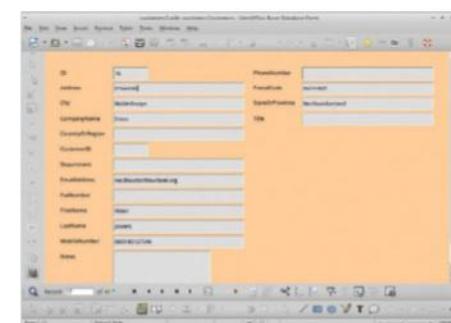
## 12 Create form from template

Select Forms from the sidebar. Click on 'Use Wizard to Create Form' in the Tasks window. In the table wizard, click on the >> symbol to copy across all of the fields in the database.



## 13 Finalise form

Accept the defaults in sections 2, 3 and 4, but select the first arrangement icon in section 5, 'Arrange controls'. You should now see a preview of our entry form in the main window. Select defaults on the other sections and then click on Finish.



## 14 Test data entry

To enter data into the database, use the form that we created. Select Forms in the sidebar and then double-click on the name of the form in the main window. This brings up the GUI record-editing interface. The form can still be tweaked and edited by right-clicking on its name in the main window.

The screenshot shows the 'Export contacts' dialog box. It has a title bar 'Export contacts'. Below it is a section titled 'Which contacts do you want to export?' with three options: 'Selected contacts (0)', 'The group My Contacts (15)', and 'All contacts (103)'. The 'My Contacts' option is selected. Underneath is another section titled 'Which export format?' with three options: 'Google CSV format (for importing into a Google account)', 'Outlook CSV format (for importing into Outlook or another application)', and 'vCard format (for importing into Apple Address Book or another application)'. The 'vCard format' option is selected. At the bottom are three buttons: 'Export' (blue), 'Cancel', and 'Learn more'.

## 15 Export your contacts from Gmail

Switch from the Gmail contacts view using the pull-down menu in the top-left corner, underneath the Gmail logo. From here, click on the More icon pull-down menu and select Export... Click on Export.

## 16 Clean up the data and create a key

Start a new spreadsheet and open the CSV file that you exported from Gmail. Use Ctrl-mousewheel zooming to get an overview. Typically, a lot of the fields will be completely blank, so select these columns (click on the column letter at the top of the window) and remove them (Edit>Delete cells). We have to create a key for each record. Label a column ID. Select the first cell in the column and then select the final cell by Shift-clicking on it. Use the fill feature (Edit>Fill>Series).

The screenshot shows the 'Fill Series' dialog box. It has sections for 'Direction' (Down, Right, Up, Left), 'Series type' (Linear, Growth, Date, AutoFill), 'Time unit' (Day, Weekday, Month, Year), and input fields for 'Start value' (300), 'End value' (empty), and 'Increment' (1). Buttons for 'OK', 'Cancel', and 'Help' are at the bottom right.

## 17 Import the data into Base

When you've cleaned up the spreadsheet, select the data (including the column headers) by clicking on the top-left cell and then Shift-clicking on the bottom-right cell. Right-click and select Copy. Select Tables from the side menu of the Base module. From here, right-click on the customers table and click on Paste. This should bring up the import wizard. Select 'Append data' and 'Use first line as column names' options, and click on Next.

The screenshot shows the 'Add user' dialog box in phpMyAdmin. It has sections for 'User information' (User name: 'User test here', Host: 'Any host', Password: 'User test here', No type: empty, Generate password: checked), 'Database for user' (Create database with same name and grant all privileges checked), and 'Add user' button.

## 18 Align the fields

The field names from our imported data don't quite match those of the database and so we need to use the second page of the wizard to line them up. To do so, click on a field name and use the up and down icons in the other list to create the correct attachments. Then click on Create.

The screenshot shows the 'Import' dialog box in phpMyAdmin. It displays 'Source table: customer' and 'Destination table: customer'. A list of fields is shown on both sides with arrows indicating mapping between them. Fields include Name, Given Name, Additional Name, Family Name, Notes, Group Membership, E-mail 1 - Type, E-mail 1 - Value, Website 1 - Value, Address, City, CompanyName, CountryOrRegion, CustomerID, Department, EmailAddress, FaxNumber, FirstName, LastName, MobileNumber, Notes, and PhoneNumber.

## 19 Create a new database user

To access the database from more than one machine, you must create additional users. Log back into phpMyAdmin, click on the Users tab and select 'Add user'. From here, create a new user with the name and password of your choosing and make a note of it. Click on 'Check all' in the 'Global privileges' section.

The screenshot shows the 'Database properties - Connection settings' dialog box. It has sections for 'General' (Name of the MySQL database: 'customer'), 'User authentication' (User name: 'mike', Password required checked), and buttons for 'Help', '<< Back', 'Next >>', 'Search', and 'Cancel'.

## 20 Redistribute the database

In the Base module, re-save the database under a new name. In this new version of the file, we have to alter a few details. Select Edit>Database>Properties and enter the name of the new database user. Click on the Additional Settings tab and enter the IP address of the machine with the MySQL database.

The replica set consists of nodes 192.168.2.4 (port 27019), 192.168.1.10 (port 27019) and 192.168.2.3 (port 27018)

Which is the primary node

```

code - mongod -- 90x41
Mon Jul 1 11:09:56.388 [rsStart] trying to contact 192.168.2.3:27018
Mon Jul 1 11:09:56.397 [rsStart] replSet I am 192.168.2.4:27019
Mon Jul 1 11:09:56.397 [rsStart] replSet got config version 1 from a remote, saving locally
Mon Jul 1 11:09:56.397 [rsStart] replSet info saving a newer config version to local.system.replset
Mon Jul 1 11:09:56.415 [rsStart] replSet saveConfigLocally done
Mon Jul 1 11:09:56.415 [rsStart] replSet STARTUP2
Mon Jul 1 11:09:56.428 [rsSync] *****
Mon Jul 1 11:09:56.433 [rsSync] creating replication oplog of size: 192MB...
Mon Jul 1 11:09:56.434 [FileAllocator] allocating new datafile ./mongo5/local.1, filling with zeroes...
Mon Jul 1 11:09:56.434 [FileAllocator] creating directory ./mongo5/_tmp
Mon Jul 1 11:09:57.730 [FileAllocator] done allocating datafile ./mongo5/local.1, size: 256MB, took 1.296 secs
Mon Jul 1 11:09:57.842 [rsSync] *****
Mon Jul 1 11:09:57.842 [rsSync] replSet initial sync pending
Mon Jul 1 11:09:57.842 [rsSync] replSet initial sync need a member to be primary or secondary to do our initial sync
Mon Jul 1 11:09:58.399 [rsHealthPoll] replSet member 192.168.1.10:27019 is up
Mon Jul 1 11:09:58.400 [rsHealthPoll] replSet member 192.168.2.3:27018 is up
Mon Jul 1 11:09:58.400 [rsHealthPoll] replSet member 192.168.2.3:27018 is now in state SECONDARY
Mon Jul 1 11:10:00.102 [initandlisten] connection accepted from 192.168.2.4:61515 #2 (2 connections now open)
Mon Jul 1 11:10:00.102 [conn2] end connection 192.168.2.4:61515 (1 connection now open)
Mon Jul 1 11:10:00.103 [initandlisten] connection accepted from 192.168.2.4:61516 #3 (2 connections now open)
Mon Jul 1 11:10:00.400 [rsHealthPoll] replset info 192.168.1.10:27019 thinks that we are down
Mon Jul 1 11:10:00.400 [rsHealthPoll] replSet member 192.168.1.10:27019 is now in state STARTUP2
Mon Jul 1 11:10:06.159 [conn1] replSet RECOVERING
Mon Jul 1 11:10:06.159 [conn1] replSet info voting yea for 192.168.2.3:27018 (3)
Mon Jul 1 11:10:06.403 [rsHealthPoll] replSet member 192.168.1.10:27019 is now in state RECOVERING
Mon Jul 1 11:10:08.415 [rsHealthPoll] replSet member 192.168.2.3:27018 is now in state PRIMARY
Mon Jul 1 11:10:13.843 [rsSync] replSet initial sync pending
Mon Jul 1 11:10:13.843 [rsSync] replSet syncing to: 192.168.2.3:27018
Mon Jul 1 11:10:14.024 [rsSync] build index local.me { _id: 1 }

```

The replica set is recovering

Synchronising data to node 192.168.2.3

# Create and save data with a MongoDB database

Forget about joins and SQL and try NoSQL databases – specifically MongoDB, the leading example

**MongoDB** is an open source document-oriented database system written in C++ by **Dwight Merriman** and **Eliot Horowitz**. It runs on UNIX machines as well as Windows and supports replication and sharding (aka horizontal partitioning) – the process of separating a single database across a cluster of machines.

Many programming languages – including C, C++, Erlang, Haskell, Perl, PHP, Python, Ruby

and Scala – support MongoDB. It is suitable for many things, including archiving, event logging, storing documents, agile development, real-time statistics and analysis, gaming, and mobile and location services.

This article will show you how to store Apache log files in a MongoDB database with the help of a small Python script, which can be found at <http://bit.ly/Kmva1v>.

## Resources

**MongoDB:** [www.mongodb.org](http://www.mongodb.org)

**Pymongo:** [api.mongodb.org/python/current/](http://api.mongodb.org/python/current/)

```
monastery:~ mtsouk$ mongo --version
MongoDB shell version: 2.4.4
monastery:~ mtsouk$ mongo
MongoDB shell version: 2.4.4
connecting to: test
> 
```

## 01 Connecting to MongoDB for the first time

Your Linux distribution probably includes a MongoDB package, so go ahead and install it. Alternatively, you can download a precompiled binary or get the source code from [www.mongodb.org](http://www.mongodb.org) and compile it yourself.

After installation, type `mongo --version` to find out the MongoDB version you are using and `mongo` to run the MongoDB shell and check if the MongoDB server process is running.

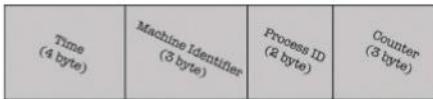
```
64.242.88.10 - - [07/Mar/2004:16:05:49 -0800] "GET /twiki/bin/edit/Main/Double
er?topicparent>Main.ConfigurationVariables HTTP/1.1" 401 12846
64.242.88.10 - - [07/Mar/2004:16:06:51 -0800] "GET /twiki/bin/rdiff/TWiki/NewU
rev1=1.3&rev2=1.2 HTTP/1.1" 200 4523
64.242.88.10 - - [07/Mar/2004:16:10:02 -0800] "GET /mailman/listinfo/hsdivision
200 6291
64.242.88.10 - - [07/Mar/2004:16:11:58 -0800] "GET /twiki/bin/view/TWiki/WikiS
1.1" 200 7352
64.242.88.10 - - [07/Mar/2004:16:20:55 -0800] "GET /twiki/bin/view/Main/DCCAnd
/1.1" 200 5253
64.242.88.10 - - [07/Mar/2004:16:23:12 -0800] "GET /twiki/bin/oops/TWiki/Appen
m?template=oopsmore&param1=1.12&param2=1.12 HTTP/1.1" 200 11382
64.242.88.10 - - [07/Mar/2004:16:24:16 -0800] "GET /twiki/bin/view/Main/PeterII
.1" 200 4924
64.242.88.10 - - [07/Mar/2004:16:29:16 -0800] "GET /twiki/bin/edit/Main/Header.
cparent>Main.ConfigurationVariables HTTP/1.1" 401 12851
64.242.88.10 - - [07/Mar/2004:16:30:29 -0800] "GET /twiki/bin/attach/Main/Offi
HTTP/1.1" 401 12851
64.242.88.10 - - [07/Mar/2004:16:31:48 -0800] "GET /twiki/bin/view/TWiki/WebTo
ate HTTP/1.1" 200 3732
```

SQL Term	MongoDB Term
Database	Database
Table	Collection
Index	Index
Row	BSON document
Column	BSON field
Primary Key	<code>_id</code> field
Group by	Aggregation
Join	Embedding and Linking

## 02 MongoDB terminology

NoSQL databases are designed for the web and do not support joins, complex transactions and other features of the SQL language. You can update a MongoDB database schema without downtime, but you should design your MongoDB database without joins in mind.

Their terminology is a little different from the terminology of relational databases and you should familiarise yourself with it.



## 03 The `_id` field

Every time you insert a BSON document in MongoDB, MongoDB automatically generates a new field called `_id`. The `_id` field acts as the primary key and is always 12 bytes long. To find the creation time of the object with `_id` '51cb590584919759671e4687', execute the following command from the MongoDB shell:

```
> ObjectId("51cb590584919759671e4687")
1e4687").getTimestamp()
ISODate("2013-06-26T21:11:33Z")
```

**Note:** You should remember that queries are case-sensitive.

## 04 Inserting an Apache log file into MongoDB

Now that you know some things about MongoDB, it is time to do something interesting and useful. A log file from Apache will be inserted inside a MongoDB database using a Python script.

The Python script is executed as follows:

```
$ zcat www6.ex000704.log.gz | python2.7 storeDB.py
```

...where `www6.ex000704.log.gz` is the name of the compressed (for saving disk space) log file.

```
1 # Programmer: Mihalis Tsoukalos
2 # Date: Wednesday 26 June 2013
3 #
4 # Description: This Python script reads an Apache log file,
5 #                 parses it and stores it in a MongoDB database
6 #
7
8 import sys
9 import pymongo
10 import re
11
12 # The number of BSON documents written
13 total = 0
14
15 # Open the MongoDB connection
16 connMongo = pymongo.Connection('mongodb://localhost:27017')
17 # Connect to database named LUD (Linux User Developer)
18 db = connMongo.LUD
19 # Select the collection to save the log file data
20 logs = db.apacheLogs
21
22 # Read the file from stdin, line by line
23 for line in sys.stdin:
24     line = line.rstrip('\n')
25     parsed = map("",join,re.findall(r'[^"]+[""]|[^"]+',line))
26     # print parsed
27     total = total + 1
28     # Construct the log entry to be inserted
29     log = [
30         {'host': parsed[0],
31          'date': parsed[1],
32          'document': parsed[4],
33          'StatusCode': parsed[5],
34          'size': parsed[6]
35      }
36
37     # Store it!
38     log._id = logs.insert(log)
39     print "The _id of the inserted post is", log._id
40
41     # Close the MongoDB connection
42     connMongo.close()
43
44 # Present the total number of BSON documents written
45 print "Total number of documents written:", total
```

## "MongoDB is supported by many programming languages"

## 05 The `storeDB.py` Python script

The `storeDB.py` script uses the `PyMongo` Python module to connect to MongoDB. The MongoDB server is running on localhost and listens to port 27017. For every inserted BSON document, its `_id` field is printed on screen. Finally, the script prints the total number of documents inserted in the MongoDB database.

The host and its port number are hard-coded inside the script, so change them to match yours.

## 06 Connecting to MongoDB using PyMongo

You first need to connect to MongoDB using:

```
connMongo =
pymongo.Connection('mongodb://
localhost:27017')
```

You then select the database name you want (LUD) using the following line of code:

```
db = connMongo.LUD
```

And finally you select the name of the collection (apacheLogs) to store the data:

```
logs = db.apacheLogs
```

After finishing your interaction with MongoDB you should close the connection as follows:

```
connMongo.close()
```

```
> db.apacheLogs.find()
{ "_id" : ObjectId("51cb590584919759671e4674"), "StatusCode" : "401", "document" : "wiki/bin/edit/Main/Double_bounce_sender?topicparent=Main.ConfigurationVariables HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:05:49 -0800", "size" : "12846" }
{ "_id" : ObjectId("51cb590584919759671e4675"), "StatusCode" : "200", "document" : "wiki/bin/rdiff/TWiki/NewUserTemplate?rev1=1.3&rev2=1.2 HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:06:51 -0800", "size" : "4523" }
{ "_id" : ObjectId("51cb590584919759671e4676"), "StatusCode" : "200", "document" : "cilmam/listinfo/hdivision HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:02:08:00", "size" : "6291" }
{ "_id" : ObjectId("51cb590584919759671e4677"), "StatusCode" : "200", "document" : "wiki/bin/view/TWiki/WikiSyntax HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/06:11:58 -0800", "size" : "7352" }
{ "_id" : ObjectId("51cb590584919759671e4678"), "StatusCode" : "200", "document" : "wiki/bin/view/Main/DCCAndPostFix HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/16:20:55 -0800", "size" : "5253" }
{ "_id" : ObjectId("51cb590584919759671e4679"), "StatusCode" : "200", "document" : "wiki/bin/oops/TWiki/AppendixFileSystem?template=oopsmore&param1=1.12&param2=1.12 HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:23:12 -0800", "size" : "11382" }
{ "_id" : ObjectId("51cb590584919759671e467a"), "StatusCode" : "200", "document" : "wiki/bin/view/Main/PeterThoeny HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/06:24:16 -0800", "size" : "4924" }
{ "_id" : ObjectId("51cb590584919759671e467b"), "StatusCode" : "401", "document" : "wiki/bin/edit/Main/Header_checks?topicparent=Main.ConfigurationVariables HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:29:16 -0800", "size" : "12851" }
{ "_id" : ObjectId("51cb590584919759671e467c"), "StatusCode" : "401", "document" : "wiki/bin/attach/Main/OfficeLocations HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:30:29 -0800", "size" : "12851" }
{ "_id" : ObjectId("51cb590584919759671e467d"), "StatusCode" : "200", "document" : "wiki/bin/view/TWiki/WebTopicEditTemplate HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:31:48 -0800", "size" : "3732" }
{ "_id" : ObjectId("51cb590584919759671e467e"), "StatusCode" : "200", "document" : "wiki/bin/view/Main/WebChanges HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:32:00 -0800", "size" : "40520" }
{ "_id" : ObjectId("51cb590584919759671e467f"), "StatusCode" : "401", "document" : "wiki/bin/edit/Main/Smtpd_etcn_restrictions?topicparent=Main.ConfigurationVariables HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:33:53 -0800", "size" : "12851" }
{ "_id" : ObjectId("51cb590584919759671e4680"), "StatusCode" : "200", "document" : "cilmam/listinfo/business HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:39 -0800", "size" : "6379" }
{ "_id" : ObjectId("51cb590584919759671e4681"), "StatusCode" : "200", "document" : "wiki/bin/edit/Main/Double_bounce_sender?topicparent=Main.ConfigurationVariables HTTP/1.1", "host" : "64.242.88.10", "date" : "07/Mar/2004:16:40 -0800", "size" : "12846" }
```

## 07 Displaying BSON documents from the apacheLogs collection

Type the following in order to connect to the MongoDB shell:

\$ mongo

Select the desired database as follows:

> use LUD

See the available collections for the LUD database as follows:

> show collections

apacheLogs

system.indexes

Lastly, execute the following command to see all the contents of the apacheLogs collection:

> db.apacheLogs.find()

If the output is long, type 'it' to go to the next screen.

## 08 A replication example

Imagine that you have your precious data on your MongoDB server and there is a

power outage. Can you access your data? Is your data safe?

To avoid such difficult questions, you can use replication to keep your data both safe and available. Replication also allows you to do maintenance tasks without downtime and have MongoDB servers in different geographical areas.

## 09 Running the three MongoDB servers from the command line

For this example, you need three MongoDB server processes running.

We ran the three MongoDB servers, on their respective machines, as follows:

\$ mongod --port 27018 --bind\_ip 192.168.1.10 --dbpath ./mongo10

--rest --replSet LUDev

\$ mongod --port 27019 --bind\_ip 192.168.2.6 --dbpath ./mongo6 --rest

--replSet LUDev

```
$ mongod --port 27018 --bind_ip 192.168.2.5 --dbpath ./mongo5 --rest --replSet LUDev
```

**Note:** You are going to see lots of output on your screen.

## 10 More information about the three MongoDB servers

You should specify the name of the replica set (LUDev) when you start the MongoDB server and have the data directory, specified by the --dbpath parameter, already created. You do not necessarily need three discrete Linux machines. You can use the same machine (IP address) as long as you are using different port numbers and directories.

```
> rs.initiate()
{
  "_id": "LUDev",
  "members": [
    {
      "_id": 1,
      "host": "192.168.1.10:27018"
    },
    {
      "_id": 2,
      "host": "192.168.2.6:27019"
    },
    {
      "_id": 3,
      "host": "192.168.2.5:27018"
    }
  ]
}
{
  "info": "Config now saved locally. Should control the replicaset"
}
{
  "ok": 1
}
>
```

## 11 The rs.initiate() command

Once you have your MongoDB server processes up and running, you should run the rs.initiate() command to actually create and enable the replica set.

If everything is okay, you will see similar output on your screen. If the MongoDB server processes are successfully running, most errors come from misspelled IPs or port numbers. The rs.initiate() command is simple but has a huge impact!

## 12 Information about replication

- Any node can be primary, but only one node can be primary at a given time.
- All write operations are executed at the primary node.
- Read operations go to primary and optionally to a secondary node.
- MongoDB performs automatic failover.
- MongoDB performs automatic recovery.
- Replication is not a substitute for backup, so you should not forget to take backups.

# “Replication is not a substitute for backup”

```
Mon Jul 1 11:09:56.388 [root@ctrl] trying to contact 192.168.2.3:29916  
Mon Jul 1 11:09:56.397 [root@ctrl] repiset 1 to 192.168.2.4:29935  
Mon Jul 1 11:09:56.397 [root@ctrl] repiset got config version 1 from a remote, saving local  
copy  
Mon Jul 1 11:09:56.397 [root@ctrl] repiset info saving a newer config version to local.sys  
on repiset  
Mon Jul 1 11:09:56.415 [root@ctrl] repiset saveconfigLocally done  
Mon Jul 1 11:09:56.415 [root@ctrl] repiset 192.168.2.3:29916  
Mon Jul 1 11:09:56.433 [root@ctrl] repiset creating replication oplog of size: 25800...  
Mon Jul 1 11:09:56.434 [filelocation] allocating new datafile ..\mngs\local.1, fitting  
with zeroes...  
Mon Jul 1 11:09:56.434 [filelocation] creating directory ..\mngs\l...tp  
Mon Jul 1 11:09:56.434 [filelocation] done allocating datafile ..\mngs\local.1, size: 2  
bytes  
Mon Jul 1 11:09:57.062 [rsync] *****  
Mon Jul 1 11:09:57.062 [rsync] repiset initial sync pending  
Mon Jul 1 11:09:57.062 [rsync] repiset initial sync need a member to be primary or seem  
primary  
Mon Jul 1 11:09:58.399 [rsuthold@ctrl] repiset number 192.168.1.1:30209 is up  
Mon Jul 1 11:09:58.399 [rsuthold@ctrl] repiset number 192.168.2.3:32903 is up  
Mon Jul 1 11:09:58.400 [rsuthold@ctrl] repiset number 192.168.2.3:32904 is now in state SE  
COMMITTING  
Mon Jul 1 11:10:00.102 [inbandlisten] connection accepted from 192.168.2.4:61654 (2 c  
onnections now open)  
Mon Jul 1 11:10:00.102 [comz] and connection 192.168.2.4:61655 (1 connection now open)  
Mon Jul 1 11:10:00.103 [inbandlisten] connection accepted from 192.168.2.4:61656 (2 c  
onnections now open)  
Mon Jul 1 11:10:00.103 [rsuthold@ctrl] repiset info 192.168.1.1:30209 thinks that we are  
now down  
Mon Jul 1 11:10:00.400 [rsuthold@ctrl] repiset number 192.168.1.1:30208 is now in state S  
TAKING  
Mon Jul 1 11:10:00.400 [rsuthold@ctrl] repiset 1 to 192.168.2.3:29916  
Mon Jul 1 11:10:00.456 [rsync] repiset RECHECKING  
Mon Jul 1 11:10:00.456 [rsync] repiset info valid poll for 192.168.2.3:29916 (X)  
Mon Jul 1 11:10:00.405 [rsuthold@ctrl] repiset number 192.168.1.1:30208 is now in state E  
CHECKING  
Mon Jul 1 11:10:00.405 [rsuthold@ctrl] repiset number 192.168.2.3:29916 is now in state PR  
Mon Jul 1 11:10:15.043 [rsuthold@ctrl] repiset initial sync pending  
Mon Jul 1 11:10:15.085 [rsync] repiset syncing to 192.168.2.3:29916  
Mon Jul 1 11:10:15.085 [rsync] build index (local= (1 : 1)
```

## 13 More information about replication

- 13 • The former primary will rejoin the set as a secondary if it recovers.

  - Every node contacts the other nodes every few seconds to make sure that everything is okay.
  - It is advised to read from the primary node as it is the only one that contains the latest information for sure.
  - All the machines of a replica set must be equally powerful in order to handle the full load of the MongoDB database.

```

    "status": {
      "set": "100ms",
      "date": 1500000000000,
      "myState": 1,
      "members": [
        {
          "id": 1,
          "name": "192.168.1.10:27018",
          "health": 1,
          "state": 1,
          "stateStr": "PRIMARY",
          "uptime": 98,
          "optime": {
            "ts": 1372357413,
            "v": 1
          },
          "syncDate": 1500000000000,
          "self": true
        },
        {
          "id": 2,
          "name": "192.168.2.6:27019",
          "health": 1,
          "state": 2,
          "stateStr": "SECONDARY",
          "uptime": 96,
          "optime": {
            "ts": 1372357413,
            "v": 1
          },
          "syncDate": 1500000000000,
          "lastHeartbeat": 1500000000000,
          "lastHeartbeatReco": 1500000000000,
          "pingMs": 0,
          "syncing": "192.168.1.10:27018"
        },
        {
          "id": 3,
          "name": "192.168.2.5:27018",
          "health": 1,
          "state": 2,
          "stateStr": "SECONDARY",
          "uptime": 96,
          "optime": {
            "ts": 1372357413,
            "v": 1
          },
          "syncDate": 1500000000000,
          "lastHeartbeat": 1500000000000,
          "lastHeartbeatReco": 1500000000000,
          "pingMs": 12,
          "syncing": "192.168.1.10:27018"
        }
      ],
      "ok": 1
    }
  }
}

```

## 14. The rs.status() command output

**14** The `rs.status()` command shows you the current status of your replica set. It is the first command to execute to find out what is going on.

```
monastery:code mtsouk$ mongo 192.168.1.10:27019
MongoDB shell version: 2.4.4
connecting to: 192.168.1.10:27019/test
Server has startup warnings:
Mon Jul 1 12:36:23.056 [initandlisten]
Mon Jul 1 12:36:23.056 [initandlisten] ** WARNING
56, should be at least 1000
> db.echo.insert({say: "Am I allowed to write?"})
not master
> 
```

Apart from primary and secondary nodes, a third type of node exists. It is called **arbiter**. An arbiter node does not have a copy of the data and cannot become primary. Arbiter nodes are only used for voting in elections for a primary node.

```

    "secondary": true,
    "replica": "15000etcd2013-06-27T18:29:41Z",
    "syncing": "192.168.2.5:27018",
    "members": [
      {
        "id": 1,
        "name": "192.168.1.10:27018",
        "health": 0,
        "state": 0,
        "status": "Not reachable/healthy",
        "version": 0,
        "optime": {
          "log": 1372397620,
          "index": 1
        },
        "lastIndex": 15000etcd2013-06-27T18:27:06Z,
        "lastTerm": 15000etcd2013-06-27T18:29:40Z,
        "lastHeartbeatIndex": 15000etcd2013-06-01T00:00:00Z,
        "prevLog": 0
      },
      {
        "id": 2,
        "name": "192.168.2.6:27018",
        "health": 1,
        "state": 2,
        "status": "SECONDARY",
        "version": 961,
        "optime": {
          "log": 1372397620,
          "index": 1
        },
        "lastIndex": 15000etcd2013-06-27T18:27:00Z,
        "lastTerm": "Syncing to: 192.168.2.5:27018",
        "lastHeartbeatIndex": 15000etcd2013-06-01T00:00:00Z,
        "prevLog": 0
      },
      {
        "id": 3,
        "name": "192.168.2.5:27018",
        "health": 1,
        "state": 1,
        "status": "PRIMARY",
        "version": 949,
        "optime": {
          "log": 1372397620,
          "index": 1
        },
        "lastIndex": 15000etcd2013-06-27T18:27:00Z,
        "lastTerm": 15000etcd2013-06-27T18:29:41Z,
        "lastHeartbeatIndex": 15000etcd2013-06-01T00:00:00Z,
        "prevLog": 1
      }
    ],
    "syncing": 192.168.1.10:27018"
  }
}

```

## 15 Selecting a new primary node

**15** If you shut down the primary MongoDB server (by pressing `Ctrl+C`), the logs of the remaining two MongoDB servers will show the failure of the `192.168.1.10:27018` MongoDB server:

Mon Jul 1 11:21:29.371 [rsHealthPoll] couldn't connect to 192.168.1.10:27018: couldn't connect to server 192.168.1.10:27018

Mon Jul 1 11:21:29.371 [rsHealthPoll] couldn't connect to 192.168.1.10:27018: couldn't connect to server 192.168.1.10:27018

It takes about 30 seconds for the new primary server to come up and the new status can be seen by running the `rs.status()` command.

**Important note:** Once a primary node is down, you need more than 50 per cent of the remaining nodes in order to select a new primary server.

## 16 Trying to write data to a non-master node

If you try to write to a non-master node, MongoDB will not allow you and will generate an error message.

17 Useful MongoDB commands

- Delete the full apacheLogs collection:  
`db.apacheLogs.drop()`
  - Show available databases: `show dbs`
  - Find documents within the apacheLogs collection that have a StatusCode of 404:  
`db.apacheLogs.find({"StatusCode": "404"})`
  - Connect to the 192.168.1.10 server using port number 27017: `mongo 192.168.1.10:27017`

**19** Hints and tips

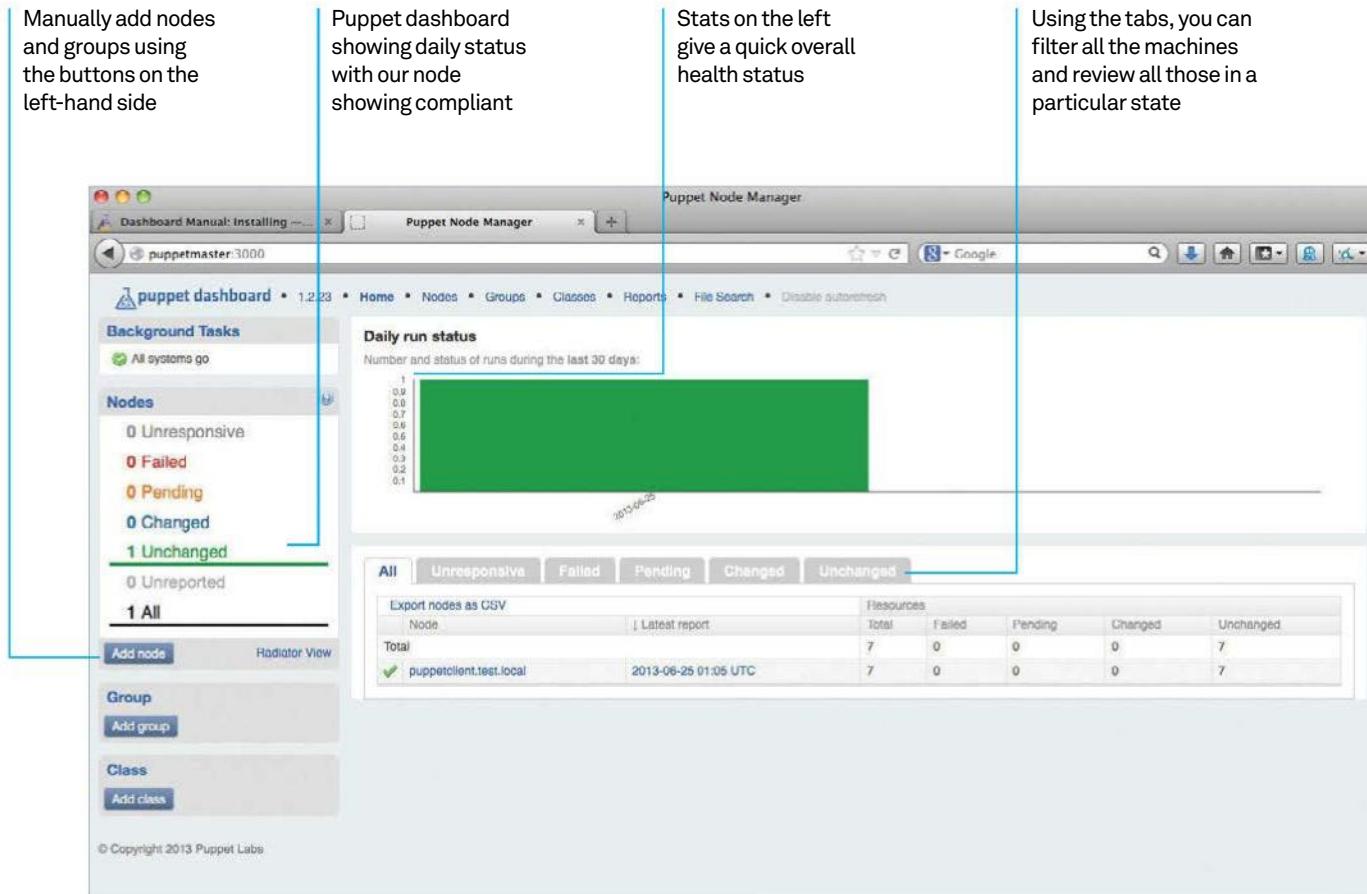
• It is highly recommended that you first run `find()` to verify your criteria before actually deleting the data with `remove()`.

- Should you need to change the database schema and add another field, MongoDB will not complain and will do it for you without any problems or downtime.

- The way to handle very large datasets is through sharding.

- Mongo has its own distributed file system called GridFS.

- The name Mongo comes from ‘humongous’.



# Maintain and manage all of your machines with Puppet

Set up your machines to be configured using Puppet so you can keep them in a consistent and workable state

## Resources

**Two networked servers**

**Ubuntu 12.04:** [www.ubuntu.com](http://www.ubuntu.com)

**Static IPs and full DNS entries**

Keeping track of two machines, keeping them in sync is quite easy – for example, **repo files and config files**. However, once you start scaling past a machine or two, keeping files aligned over tens, hundreds and thousands of machines, it becomes a nightmare. If there is one thing a network manager likes, it is configuration standards.

This is where Puppet comes in. Puppet allows users to use extend control over the

contents of their files and keep them in sync across your estate. In this how-to guide to implementing a basic Puppet setup, we show how to keep all your files in sync.

This tutorial covers the basics of creating a basic Puppet server and client setup, through to setting up a few sample configurations that can be deployed, applying different configurations to different machines and configuring to clients in a standard manner.

The image shows two terminal windows side-by-side. The left window, titled 'stu@puppetmaster: ~ - ssh - 80x24', displays the output of a Puppet master installation command. It includes dependency resolution for packages like libreadline5, libyaml, and various Ruby and Puppet components. The right window, titled 'stu@puppetclient: ~ - ssh - 80x24', shows the output of a Puppet client installation command. It includes dependency resolution for packages like libreadline5, libyaml, and various Ruby and Puppet components. Both windows show the command 'sudo touch /etc/puppet/manifests/site.pp' being run at the end.

## 01 Set up the Puppet master

Puppet comes in two parts – master and agent nodes. The master node, as the name implies, is in charge. This server holds all the config file goodness (also known as manifests). For this tutorial we are running Ubuntu 12.04 LTS. Installing Puppet is really straightforward. Choose one of the hosts and install the Puppet master. Type the command:

```
sudo apt-get install puppetmaster
```

The setup requires the file `site.pp` to be present (more on what it is later). Do this by using:

```
sudo touch /etc/puppet/manifests/
site.pp
```

This installs all the prerequisites of the server.

## 02 Set up the Puppet agent

The agents sit on the machines that we want to effectively manage. To install all the components, use the command:

```
sudo apt-get install puppet
```

Again, this installs all the requirements for the agent or client. It is suggested that you do not set Puppet to autostart on boot. If you do this, by default the agent will contact the Puppet master and update its configuration, if needed, every 30 minutes. We are going to run ours manually, so that there is no waiting to see the changes take effect.

## 03 Configure the Puppet infrastructure

The next step is to set up the secure communication between the servers. To do this, log into the Puppet agent server and issue the command:

```
sudo puppetd --server puppetmaster.
test.local --waitforcert 60 --test
```

You will have to edit the server name to reflect your setup. Leave the `--test` switch on as it'll show what is happening in the foreground, making life easier if there's a need to debug.

If you run the command and you get an error 'warning: Could not retrieve fact fqdn', it means you have not set up your DNS properly. It is strongly recommended that this is fixed before proceeding.

The image shows a terminal window titled 'stu@puppetmaster: ~ - ssh - 80x24'. It displays the command 'sudo puppetca --list' followed by 'stu@puppetmaster:~\$ sudo puppetca --sign puppetclient.test.local'. The output shows a signed certificate request for 'puppetclient.test.local' with a fingerprint of '56:10:CG:02:2E:2D:25:39:3E:E9:67:E3'.

## 04 It's good to talk SSL

The next step is to enable secure communications between the master and agent.

Type `sudo puppetca --list`. This will show all the client machines that are trying to connect to the server to service their requests. In order for them to be given access, we must allow them to do so, using the command:

```
sudo puppet cert --sign clientname
```

Look at the agent console while doing this and see the handshake that is going on as the machines are joined together.

To test if an agent system can see the server, there is a command that can be used to test.

## 05 Introducing some Puppet basics

Before all the interesting code creation, you need to understand how Puppet works. All the configurations are held in manifest files. Manifests are just source files are what we can edit. All source files end in `.pp`

The whole point of having a Puppet setup is to ensure that the machines on the site are all the same (we can differentiate between server

types later!). To help with this aim, Puppet thoughtfully created a site-wide basic config file called `site.pp`. This is a basic file that is used to create the configurations.

To create changes on systems, a manifest is used. A manifest is a number of (or just one) text files. Within these manifests are the details that configure each part of the system that can be edited and customised. Looking at a very, very basic manifest – it is fairly clear as what it does...

```
file {'myfile':
  path    => '/tmp/myfile',
  ensure  => present,
  mode    => 0640,
  content => "This could be
anything.",}
```

The first line is termed a resource. Resources are groups of similar things that can be configured to meet a desired standard. Examples of resources include directories, services and files. In other words, basically groups of items that share a commonality.

The bit after the file resource is what is known as the title. It can be thought of as the unique identifier. The bits that follow the identifier are properties and values. To explain it a bit better, the resource 'file' has a number of properties, such as the path and the file rights.

The image shows a terminal window titled 'stu@puppet...ests: ~ - ssh - 80x24'. It displays a Puppet manifest file named 'site.pp' with the following content:  

```
file {'myfile':
  path    => '/tmp/myfile',
  ensure  => present,
  mode    => 0640,
  content => "This could be anything.",}
```

```
stu@puppet...ifests - ssh stu@puppetclient: ~ - ssh
stu@puppetmaster:/etc/puppet/manifests$ sudo puppet apply test.pp
notice: /Stage[main]/File[myfile]/ensure: created
notice: Finished catalog run in 0.16 seconds
stu@puppetmaster:/etc/puppet/manifests$
```

## 06 Testing the Puppet manifests

Manifests can be tested on the local Puppet master machine if you want to (not best practice, but will suffice for the tutorial needs). Simply save the above into a file, for example **test.pp**. Once you have done that, use the command:

```
sudo puppet apply /path/to/test.php
```

One item by itself is not very useful, so we could group together several items in one file. However, it makes more sense to split down the manifests into the jobs they do – or, to use the proper term, classes. That way you can modify the manifests to meet the requirements for multiple groups.

```
stu@puppet...ifests - ssh stu@pup
package { "apache2":
    ensure => "present"
}

service { "apache2":
    enable => "present"
}

~
~
```

## 07 Doing useful stuff with the manifest

It was mentioned before that we could do useful things with Puppet. For example, it can be made to install an application. This can be done by defining the Resource; this time the resource is ‘package’ and using the `ensure` property followed by ‘`ensure`’ to make sure it is installed or

**“The whole point of having a Puppet setup is to ensure that the machines on the site are all the same”**

```
'absent' to make sure it is not!
package { "apache2":
    ensure => "present"
}
```

With a simple addition, that basic start can be built up to autostart. The resource this time is ‘service’. Following on from the above, add:

```
service { "apache2":
    enable => 'true'
}
```

```
stu@puppet...asses - ssh stu@p
class webserver {
    package { "apache2":
        ensure => "present"
    }

    package { "php5":
        ensure => "present";
    }

    service { "apache2":
        enable => 'true'
    }
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"webserver.pp" 14L, 159C
```

## 08 Doing it cleanly with classes

Putting all these entries in one file is going to get messy, right? Also what if there are several different configurations? This is where the system can be used to differentiate. Use classes to group together bits of code that need to run, but reference it rather than putting all the code directly into `site.pp`.

If there was a need for a separate web server config and a database configuration, they’ll have some commonalities and some differences.

So create a folder called `classes` under the manifest folder.

Create a new file under `classes`, call it `webserver.pp` and put in the following:

```
class webserver {
    package { "apache2":
        ensure => "present";
    }

    package { "php5":
        ensure => "present";
    }

    service { "apache2":
        enable => 'true'
    }
}
```

```
stu@puppet...ifests - ssh # /etc/puppet/manifests/site.pp
import "classes/*"

node default {
    include webserver
}
```

## 09 Making the class useful!

Once the class is created, it can be referenced in the `site.pp` file. To make it work, the classes need to be included in the latter. Go back to `site.pp` and modify it to include the following text:

```
# /etc/puppet/manifests/site.pp
import "classes/*"
node default {
    include webserver
}
```

To import the classes, we just use the `import` command. The ‘`default`’ means it is applied to all nodes. Notice how we use ‘`include webserver`’ and the class is called `webserver`? Basically, the class can be called by using ‘`include`’ suffixed by the class created that is to be referenced.

The `default` node is applied to all the nodes. It is possible to create nodes with special uses and work only on specific nodes. These nodes are the same layout as the `default`, except they have different names. Again, the `include` can be used to apply several configurations to all new nodes.

## 10 Assembling all the parts of Puppet

How do we group machines together and apply specifics? It’s quite straightforward. Use the ‘`node`’ prefix. Again it goes into `site.pp`. An example of adding specific machines is:

```
node 'puppetclient2.test.local'
inherits database
```

...stating the node with its fully qualified name, and stating that it inherits, or takes on, the setup that is contained in the database class. In this example, puppetclient2 would get all the database configurations defined in the following way:

```
node database {
  include mysql-server
}
```

Then, when we have a database server, it will read all the classes defined in the node. So, for example, the database nodes get all the class setups contained within MySQL, Apache and phpMyAdmin with their potentially customised configuration. So we can essentially chain together several classes into one node. Note, though, that the default class must always come first!

```
# config/database.yml

database: dashboard
username: dashboard2
password: my_password
encoding: utf8
adapter: mysql

development:
  database: dashboard_development
  username: dashboard
  password:
  encoding: utf8
  adapter: mysql

test:
  database: dashboard_test
  username: dashboard
  password:
  encoding: utf8
  adapter: mysql
```

## 11 Installing the web report interface

There is also a web management interface that can be installed to make reporting and management easier. It is not straightforward, but using it makes for a more interesting and GUI experience. Installing the Puppet dashboard on the Puppet master for evaluation is okay.

Because of the dependencies you may encounter, your life will be much easier if you install it from the Puppet Labs repository. Enabling it is quite straightforward, as is

shown below. On the Puppet master, use the following command to get the repository details:

```
wget http://apt.puppetlabs.com/
puppetlabs-release-precise.deb
```

Followed by installing the repository:

```
sudo dpkg -i puppetlabs-release-
precise.deb
sudo apt-get update
```

Now that we have the repository installed, the dashboard can be installed using the command below. It will also install all the other dependencies needed:

```
sudo apt-get install puppet-
dashboard
sudo apt-get install mysql-server
```

Now that the files are installed, you will need to set up the database. Run MySQL as root (just while we create the database and users). Each line needs to be entered one line at a time.

```
CREATE DATABASE dashboard_production
CHARACTER SET utf8;
CREATE USER 'dashboard'@'localhost'
IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON
dashboard_production.* TO
'dashboard'@'localhost';
```

Once the database is set up, take the password used above and edit the config file with the command:

```
sudo vi /etc/puppet-dashboard/
database.yml
```

You will see a series of text lines, one headed with the item 'Production'. Fill in the password field with my\_password.

## 12 Configure the data collection

Once the database is created, it needs to be populated. This is done using the following set of commands:

```
cd /usr/share/puppet-dashboard
rake RAILS_ENV=production db:migrate
sudo -u www-data /usr/share/
puppet-dashboard/script/server -e
production
```

This will load the Puppet server in the foreground. Watch the console for any errors that are thrown up. The file /etc/puppet/puppet.conf will need to be edited. On the client, edit it with these values:

```
[agent]
report=true
```

Edit /etc/puppet/puppet.conf on the master to add to the '[master]' section:

```
reports = store, http
reporturl = http://puppetmaster.
test.local:3000/reports/upload
```

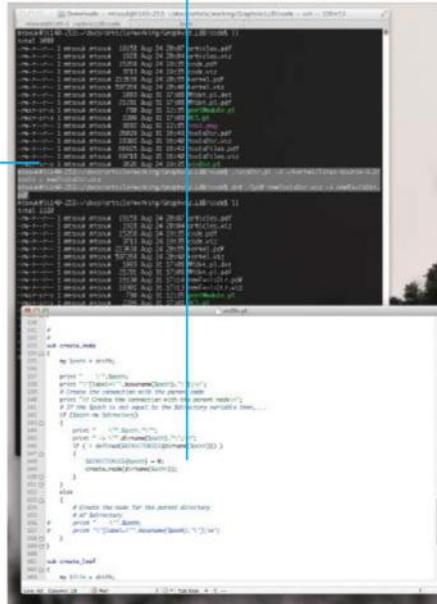
The web server can be accessed using the following URL: <http://puppet-server:3000>

Log into the client and then rerun the Puppet agent. Once this is done, if the foreground agent is run and the server is active, you will see there is a pending task. Log into the Puppet master and run the following command:

```
cd /usr/share/puppet-dashboard
sudo -u www-data env RAILS_
ENV=production script/delayed_job -p
dashboard -n 4 -m start
```

This will run a background collection script that will populate the web server with collected data. And there you have it.

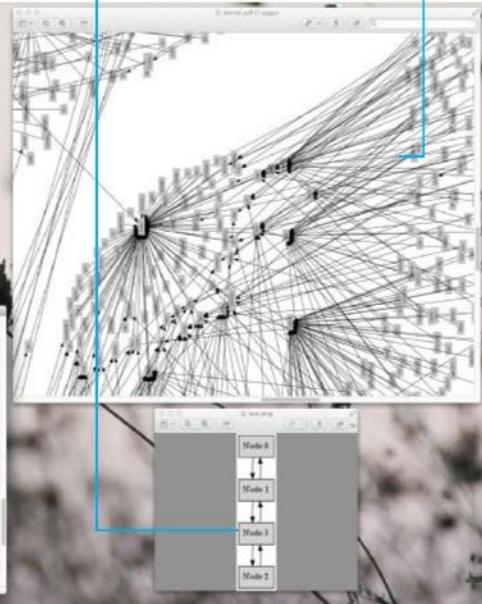
Running the visDir.pl Perl script and creating a PDF file using dot



Part of the visDir.pl Perl script

An example of a simple Graphviz graph

A small part of the Linux kernel directory structure!



# Visualise directory structures with Graphviz

Make large directory structures practical with this open source visualisation package

**Graphviz** is a collection of tools for manipulating graph structures and generating graph layouts. It supports both directed and undirected graphs and offers graphical and command-line tools – we'll be using the latter.

Graphviz contains many programs and libraries. The dot program is a utility for drawing directed graphs. It accepts input in the dot language. The dot language can define three kinds of objects: graphs, nodes and edges. Neato is a program for drawing undirected graphs, which are commonly used for telecoms and computer programming tasks. The circo utility is used for creating circular layouts of graphs, while fdp generates undirected graphs. The sfdp program is a utility for constructing large undirected graphs. The twopi program

is a utility for drawing graphs using a circular layout. One node is chosen as the centre, and the other nodes are placed around the centre in a circular pattern. If a node is connected to the centre node, it is placed at distance 1. If a node is connected to a node directly connected to the centre node, it is placed at distance 2 and so on.

Graphviz also provides three graphical programs named dotty, tcldot and lefty: dotty is a graphical editor for technical pictures; lefty is a customisable interface for the X Window System written in lefty; tcldot is a customisable graphical interface written in Tcl 7. There are also two drawing libraries called libgraph and libagraph. Their existence means that an application can use Graphviz as a library rather than as a software tool.

## Resources

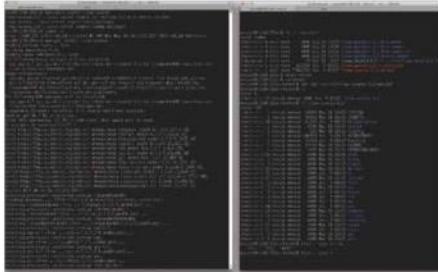
**Graphviz:** [www.graphviz.org](http://www.graphviz.org)

**A text editor**

**Perl:** [www.perl.org](http://www.perl.org)

**Perl Graphviz module:** [search.cpan.org/~lbrocard/GraphViz-2.02/lib/GraphViz.pm](http://search.cpan.org/~lbrocard/GraphViz-2.02/lib/GraphViz.pm)

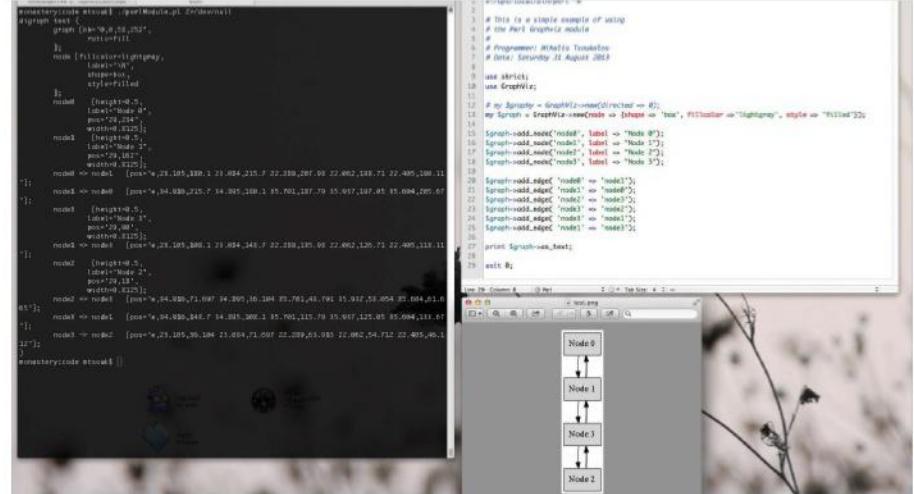
**Graphviz output formats:** [www.graphviz.org/doc/info/output.html](http://www.graphviz.org/doc/info/output.html)



## 01 Why Graphviz?

Visualising large directory structures such as the Linux kernel can be really practical. The Linux kernel root directory contains over 2,000 other directories and 37,000 files that would otherwise be very difficult to picture. The output of our Perl script can optionally show the included files as well as their sizes. Also, the Graphviz knowledge you'll get by visualising directories can be used for visualising networks, traceroute paths, function calls etc. And there are plenty of other benefits.

**Note:** For huge directory structures such as the Linux kernel, it is better not to visualise all at once but to split into smaller parts.



```
"Hello world!";
```

```
}
```

Use the following command for the compilation:

```
$ dot -Tps hw.dot -o hw.ps
```

The aforementioned command will produce a PostScript file called hw.ps that you can view. The word digraph means that a directed graph is going to be created. For creating an undirected graph, the word graph should have been used instead. For such a simplistic example, however, it does not make any difference if the graph is either directed or undirected.

Although the PostScript format used to be very popular, it is recommended to use the PDF format because it is faster to render and display. Additionally, PDF files can be zoomed in more before losing their clarity.

## 03 The Perl Graphviz module

Many programming languages, including Python, Ruby, C++ and Perl, provide their own interface for creating Graphviz files. The Perl module is presented as an alternative way of generating Graphviz code.

The important thing to remember when using the Graphviz module is that if you want to get the output as a PNG file, the last line of your program should be:

```
print $graph->as_png;
```

Similarly, if you want to get the output as plain text, you should use the following line instead:

```
print $graph->as_text;
```

## 02 Installing and running Graphviz

Your Linux distribution probably includes a ready-to-install Graphviz package that you can use. For a Debian 7 system, you just have to run the following command to download and install Graphviz:

```
# apt-get install graphviz
```

After installing Graphviz, try to compile the following Graphviz code:

```
digraph G
{
```

## 04 Basic Graphviz information

A graph  $G(V,E)$  is a finite, non-empty set of vertices  $V$  (or nodes) and a set of edges  $E$ . A graph contains nodes and edges, each of them having attributes.

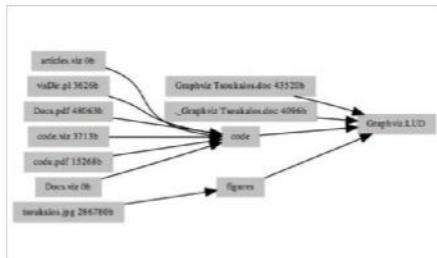
Graphviz has its own dialect that you will have to learn. The language may be simple and elegant but it is also very powerful. The good thing about Graphviz is that you can write its code using a simple plain text editor – a wonderful side effect of it is that you can easily write scripts that generate Graphviz code.

By reading some Graphviz code, you will soon realise that lines beginning with # or // are considered comments.

Node Attributes		
Name	Explanation	Allowed Values
shape	The shape of the node	ellipse, diamond, box, circle
height	The height in inches	a number
width	The width in inches	a number
label	The name of the node	alphanumeric
fontsize	The size of the font	a number
fontname	The name of the font	Courier, Helvetica, Times
fontcolor	The color of the font	white, black, blue, etc.
style	The style name	bold, dotted, filled, etc.
color	The color of the node shape	white, black, etc.
pos	The coordinates of the position	

Edge Attributes		
Name	Explanation	Allowed Values
label	The label of the edge	alphanumeric
fontsize	The size of the font	
fontname	The name of the font	
fontcolor	The color of the font	
style	The style name	bold, dotted, filled, etc.
color	The color of the edge	white, black, blue, etc.
len	The length of the edge	
dir	The direction of the edge	forward, back, both or none
decorate	Draws a line that connects labels with their edges	0 or 1
id	Optional value to denote different edges	alphanumeric

## “Experiment to find the suitable tool and parameters for the job”



### 05 A simple Graphviz example

The following Graphviz code draws a simple directory structure that includes files. It also displays the size of a file:

```
digraph Widget
{
    size="16,6";
    nodesep=0.05;
    rankdir = LR;
    rotate = 90;
    edge[len=5];
    node[style=filled, shape=record,
    fontsize=8];
    node[height=0.20, width=0.20,
    color=gray];

    "Graphviz Tsoukalos";
    doc"[label="Graphviz Tsoukalos.doc
43520b"];
    "Graphviz Tsoukalos.doc" -> "/home/
mtsouk/docs/article/working/Graphviz.LUD";
    "/home/mtsouk/docs/article/working/
Graphviz.LUD"[label="Graphviz.LUD"];

    "...Graphviz Tsoukalos.doc"[label="...
Graphviz Tsoukalos.doc 4096b"];
    "...Graphviz Tsoukalos.doc" -> "/home/
mtsouk/docs/article/working/Graphviz.LUD";
    "/home/mtsouk/docs/article/working/
Graphviz.LUD/code"[label="code"];

    "/home/mtsouk/docs/article/working/
Graphviz.LUD/code" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD";
```

```
"articles.viz"[label="articles.viz
0b"];
"articles.viz" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/code";
"visDir.pl"[label="visDir.pl 3626b"];
"visDir.pl" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/code";
"Docs.pdf"[label="Docs.pdf 48063b"];
"Docs.pdf" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/code";
"code.viz"[label="code.viz 3713b"];
"code.viz" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/code";
"code.pdf"[label="code.pdf 15268b"];
"code.pdf" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/code";
"Docs.viz"[label="Docs.viz 0b"];
"Docs.viz" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/code";
"/home/mtsouk/docs/article/working/
Graphviz.LUD/figures"[label="figures"]];

"/home/mtsouk/docs/article/working/
Graphviz.LUD/figures" -> "/home/mtsouk/
docs/article/working/Graphviz.LUD";
"tsoukalos.jpg"[label="tsoukalos.jpg
286780b"];
"tsoukalos.jpg" -> "/home/mtsouk/docs/
article/working/Graphviz.LUD/figures";
}
```

The code (articles.viz) can be compiled using the following command:

```
$ dot -Tpdf articles.viz -o articles.pdf
```

The presented visDir.pl Perl script generates similar Graphviz code.

### 06 A complicated Graphviz example

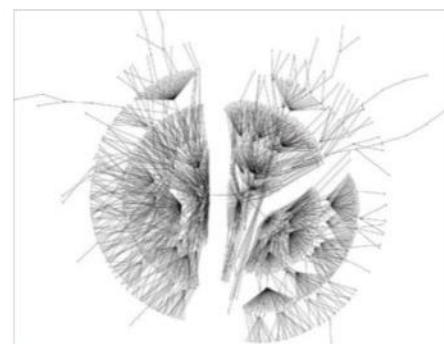
You can visualise the Linux kernel directory structure with the help of the visDir.pl script using the following two commands:

```
$ ./visDir.pl -d ~/kernel/linux-source-3.2
```

## “The Linux kernel root directory contains over 37,000 files”

```
> kernel.viz
$ neato -Tpdf kernel.viz -o kernel.pdf
```

As you already know, the Graphviz suite contains many tools for creating graphs. For graphs with a large number of nodes, you should experiment to find the suitable tool and parameters for the job. The output of the dot tool for the Linux kernel directory structure is not as pretty as the graph created using the neato tool.



### 07 The Perl script

The Perl script, called visDir.pl, requires one command-line option and one argument. The argument is the path of the directory that is going to be visualised. The command-line option must be -d (for including directories only) or -f (for also including files). If none of them is found, the script prints an explanatory message and stops execution. Please note that the directory argument must not contain a '/' at the end; so the following command will not work properly:

```
$ ./program_name.pl /usr/
```

```
35 die <<Thanatos unless @ARGV;
36 usage:
37 $0 [-f|-d] directory
38 Thanatos
39
40 if (@ARGV != 2)
41 {
42     die <<Thanatos
43     usage info:
44     Please use exactly 2 arguments!
45     Use -F if you want to include files
46     Use -d if you want to include directories ONLY
47     Thanatos
48 }
49
50 # Get the File name
51 ($nofiles,$directory) = @ARGV;
52
53 # print $nofiles, "\n";
54
55 # Check the $nofiles variable
56 if ($nofiles eq "-d")
57 {
58     $nofiles = 1;
59 }
60 elsif ($nofiles eq "-f")
61 {
62     $nofiles = 0;
63 }
64 else
65 {
66     die <<Thanatos
67     Use -F if you want to include files
68     Use -d if you want to include directories ONLY
69     Thanatos
70 }
```

```

72 print <<START;
73 digraph Widget
74 {
75     size="16,6";
76     nodesep=0.05;
77     rankdir = LR;
78     rotate = 90;
79     edge[plen=5];
80     node[style=filled, shape=record, fontsize=8];
81     node[height=0.20, width=0.20, color=gray];
82
83 START

```

## 08 Explaining the Perl script – part 1

The first thing that the Perl script prints is the common attributes of all the nodes. If you want to make global changes to the output, you should look at this part of the script.

Depending on the total number of nodes of your own graph, you may need to experiment with the given part of the script to beautify the output. The ‘rankdir = LR’ command dictates that graph nodes must be drawn from left to right. The ‘rotate = 90’ command draws the output in landscape mode.

```

95 sub create_graphviz
96 {
97     print $_;
98     # print "\n";
99     # Skip ., .., .DS_Store and ALL png files
100    # You can change it to exclude the files you want
101    if ($_ =~ /(^|\.)(\..*|DS_Store|png)$/) {
102        # do nothing!
103    }
104    else {
105        # If it is a directory, then...
106        if (-d $file::Find::name) {
107            # Duplicates can only exist in directories.
108            # We must take care of it.
109            if (!defined($DIRECTORIES{$file::Find::name})) {
110                $DIRECTORIES{$file::Find::name} = 0;
111                create_node($file::Find::name);
112            }
113        }
114        # If you do not want to display files
115        elsif ($nofiles) {
116            # do nothing!
117        }
118        # It is a file, then...
119        else {
120            create_leaf($basename($file::Find::name));
121        }
122    }
123 }
124
125
126
127
128
129 }

161 sub create_leaf
162 {
163     my $file = shift;
164     my $size = 0;
165
166     # It is always a good idea to check twice!
167     if (-f $file)
168     {
169         # This finds the size of the file in bytes
170         $size = -s $file;
171     }
172     # add the byte symbol at the end of the byte number
173     $size .= "b";
174
175     # Create the file node
176     print "    \"$file";
177     print "[label=\"$basename($file).\".\"$size.\"];\n";
178     print "    \"$file.\";\n";
179     print "    \"$dirname($file::Find::name).\";\n";
180     if (!defined($DIRECTORIES{$dirname($file::Find::name)})) {
181
182         $DIRECTORIES{$dirname($file::Find::name)} = 0;
183         create_node($dirname($file::Find::name));
184     }
185 }

```

## 09 Explaining the Perl script – part 2

The create\_graphviz function does most of the work. First, it excludes files or

directories that match given patterns, using the following command:

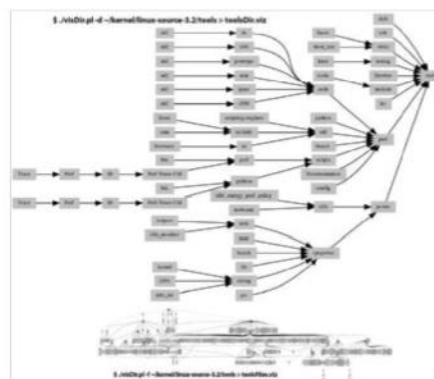
```

if ( $_ =~ /(^|\.)(\..*|DS_Store|png)$/) {
    || $_ =~ /(^|\.)(\..*|DS_Store|png)$/ )

```

Second, every time a directory is found, it calls the create\_node function to construct a new node for it.

Third, if it has to process a file, it calls the create\_leaf function (files cannot include other files so they are considered the leaves of a graph). The create\_leaf function not only produces a node for the given file, but also finds out the file size and includes it in the output.



## 10 Explaining the Perl script – part 3

The following Perl code checks if the -d option is given in order to exclude all files from the output:

```

if ( $nofiles eq "-d" )
{
    $nofiles = 1;
}

```

Should you want to exclude Word documents (their file extension is either doc or docx) from the output but include PNG files, you should use the following code:

```

if ( $_ =~ /(^|\.)(\..*|DS_Store|doc|docx)$/ ) {
    || $_ =~ /(^|\.)(\..*|DS_Store|png)$/ )

```

**Note:** The first regular expression (/^\.?\\$/) excludes ‘.’ and ‘..’ directories from the output. ‘.DS\_Store’ files are also excluded from the output as they are Mac OS X specific and are used for storing custom directory attributes.

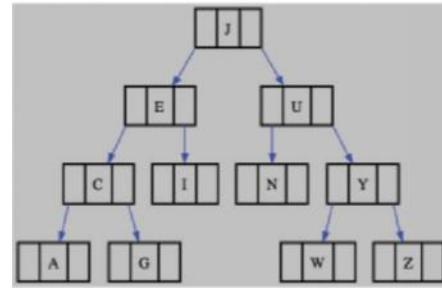
As far as the Linux kernel source is concerned, it would be a good idea to exclude Makefile and Kconfig files as well as object files from the output in order to simplify it. Excluding Makefile

and Kconfig files is as easy as using the following line of code:

```

if ( $_ =~ /(^|\.)(\..*|Makefile|Kconfig)$/ ) {
    || $_ =~ /(^|\.)(\..*|DS_Store|obj)$/ )

```



## 11 More about Graphviz

Graphviz can also be used for creating database relation diagrams between tables, UML diagrams, website maps, source code diagrams, computer network diagrams, binary trees, hash tables and knowledge representation diagrams. The good thing is that Graphviz algorithmically arranges the graph nodes so that the output is both practical and pleasing!

```

1 digraph G
2 {
3     graph [bgcolor=gray];
4     node [style=bold, label=~"N", shape=record];
5     edge [color=blue];
6     graph [bb="0,0,399,252"];
7     node0 [label=<f0> | <f1> J | <f2> " , width="0.83", height="0.59"];
8     node1 [label=<f0> | <f1> E | <f2> " , width="0.89", height="0.59"];
9     node2 [label=<f0> | <f1> C | <f2> " , width="0.89", height="0.59"];
10    node3 [label=<f0> | <f1> I | <f2> " , width="0.89", height="0.59"];
11    node4 [label=<f0> | <f1> U | <f2> " , width="0.89", height="0.59"];
12    node5 [label=<f0> | <f1> N | <f2> " , width="0.82", height="0.59"];
13    node6 [label=<f0> | <f1> Y | <f2> " , width="0.82", height="0.59"];
14    node7 [label=<f0> | <f1> W | <f2> " , width="0.84", height="0.59"];
15    node8 [label=<f0> | <f1> Z | <f2> " , width="0.89", height="0.59"];
16    node9 [label=<f0> | <f1> A | <f2> " , height="0.59"];
17    node10 [label=<f0> | <f1> G | <f2> " , height="0.59"];
18    node0:f0 --> node1:f1;
19    node0:f2 --> node2:f1;
20    node1:f0 --> node4:f1;
21    node1:f2 --> node6:f1;
22    node2:f0 --> node5:f1;
23    node4:f0 --> node7:f1;
24    node4:f2 --> node8:f1;
25    node5:f0 --> node9:f1;
26    node6:f0 --> node10:f1;
27    node9:f0 --> node8:f1;
28    node9:f2 --> node10:f1;

```

## 12 Bonus: Using Graphviz to create binary trees

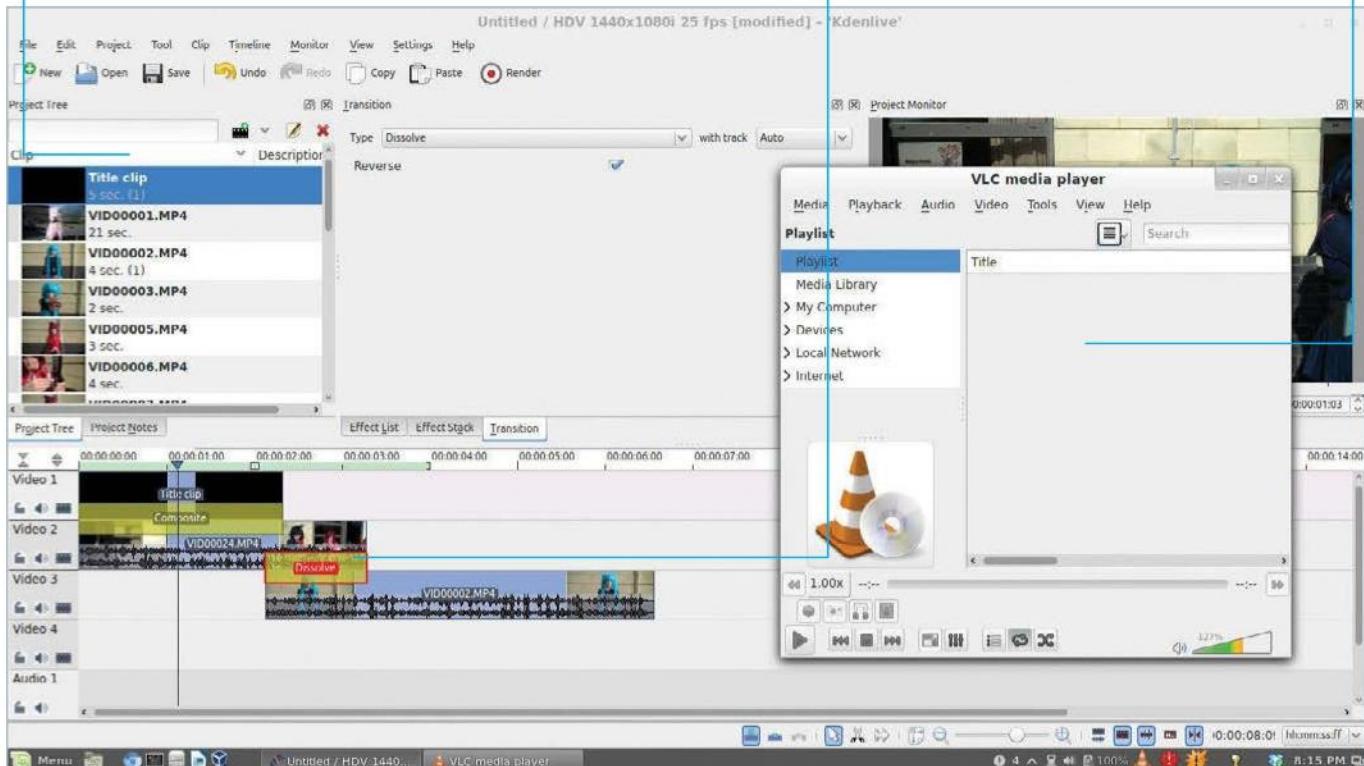
It is very easy to create a binary tree representation using Graphviz and the dot language – it would have been a lot harder to illustrate it in either Adobe Illustrator or Adobe Photoshop. Better yet, it is also easier to make changes to it. You can even search and replace, as it is just a plain text file!

Each node is divided into three parts. Each part has a name: <f0> for the first part, <f1> for the second part and <f2> for the third part. In order to call a given part of a node, the notation is node0:f0 – for the first part of node 0. The symbolic name has nothing to do with the displayed label. Also, a node part can be empty but still have a symbolic name.

Learn how to manage your clips and project settings from within the Kdenlive interface

Use transitions and effects to move between clips and tracks to create great-looking videos

Export your videos to play anywhere in a whole host of formats, thanks to the powerful encoding tools



# Edit video in Kdenlive

Create great-looking videos with readily available open source software

**One of the fields for which Linux is regularly overlooked is its media capabilities.** Non-free codecs and questionable legalities aside, Linux works with a wide variety of popular and obscure media files and has done for a long time. This also extends to its video and audio editing capabilities, served by a selection of excellent apps for many different skill levels.

Kdenlive is one of our favourites and an excellent place to jump off on. It's easy to use yet comes with a wide variety of prosumer-level features. This month we'll start by looking at some of the basic editing tricks you can use.



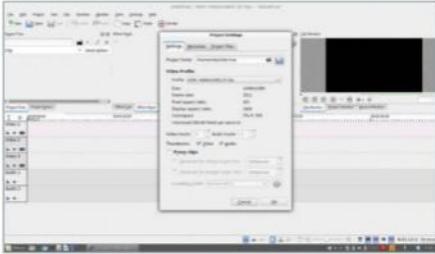
## 01 Getting Kdenlive

Kdenlive is available in the repos of many systems, so searching the software or package manager is a good place to start. Otherwise, you can live-boot or install AV Linux, which is an editing-focused distro with Kdenlive pre-installed. The latter is better if you plan to do this a lot.

## Resources

**Kdenlive** [www.kdenlive.org](http://www.kdenlive.org)

**AV Linux** [www.bandshed.net/AVLinux.html](http://www.bandshed.net/AVLinux.html)

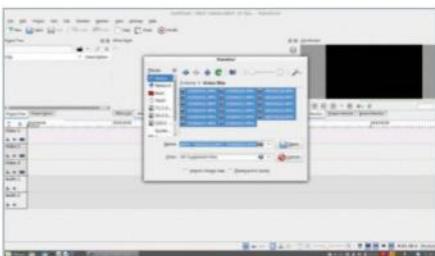


## 02 New project

Open up Kdenlive and you'll be presented with the basic interface. It will also default to specific video settings which will determine the way the final product is rendered, and how the different clips will relate to each other if they're of different dimensions. Go to Project and then Project Settings to change this.

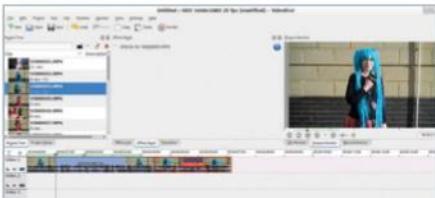
## 03 Video settings

You can choose from a wide selection of standard formats varying in resolution and frame rate. If you're not sure what size you want your video to be, the first clip you import onto the timeline can also be used as a guideline for how you want the final product to look.



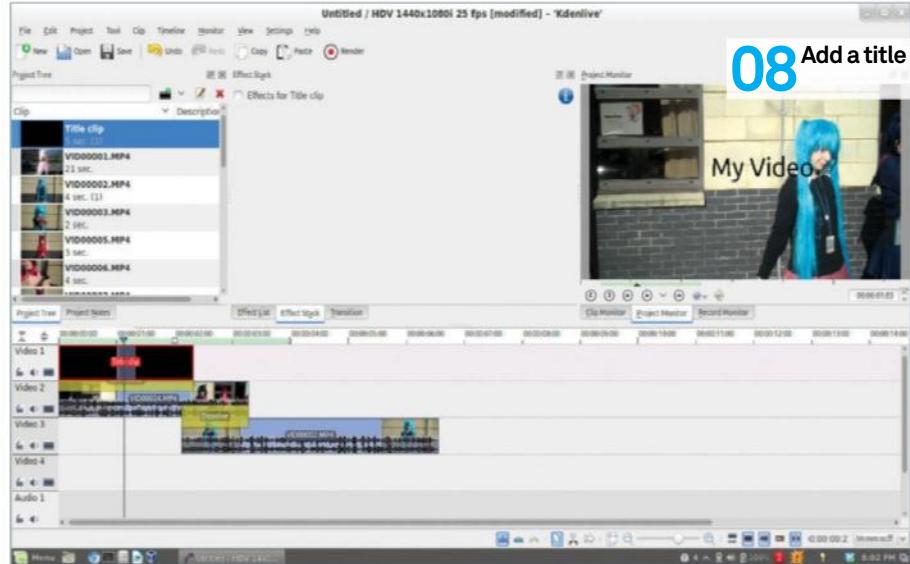
## 04 Import video

Go to Edit>Add Clip (or click on the film-strip symbol with a '+' icon) to add files. You can also add music or other audio files this way, and we'll discuss those later. Select multiple clips at a time or just the one you need right now.

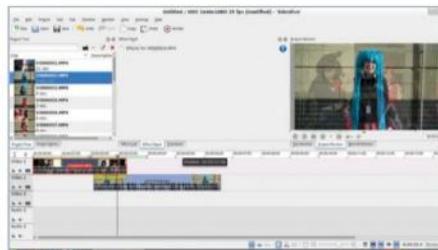


## 05 Add to timeline

Drag clips to the timeline to arrange them as you wish. You can drag them up and down to make space before or after. The highest timeline on the interface will take priority for video, which is important for fading and moving between clips.



**"Kdenlive is one of our favourite video editors, it's easy to use yet comes with a wide variety of prosumer-level features for creating great-looking videos"**

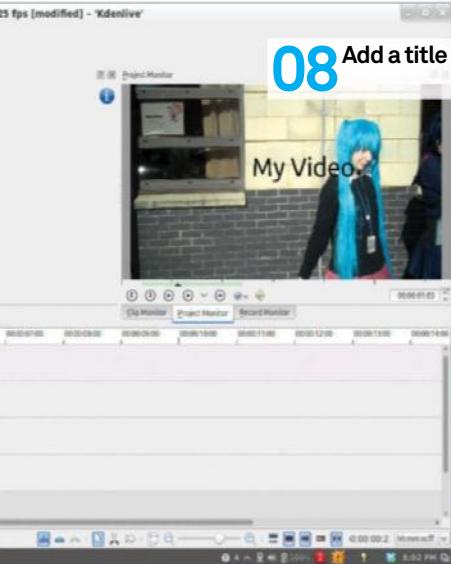


## 06 Transitioning between clips

Place the clip you want to transition to below the original clip. Right-click on the original clip and find the Add Transition menu – in there is the full list of transition types you can use. Play around with them as you wish; one tip we have is adding a pure black image if you want to fade to black.

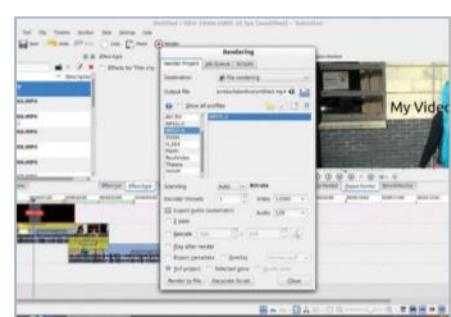
## 07 Add music

Music can be added like any of the clips: you just need to add them to an Audio track on the interface. Drag and position them like you did the video clip – if you need to edit the volume of either, right-click and go to Add Effect>Audio Correction and Volume.



## 08 Add a title

Go to Project>Add Title Clip to create a graphic title for your project. This can be text or images as you desire and is set by default to be transparent – once it's saved, add it to your timeline above the timeframe you want it to appear.



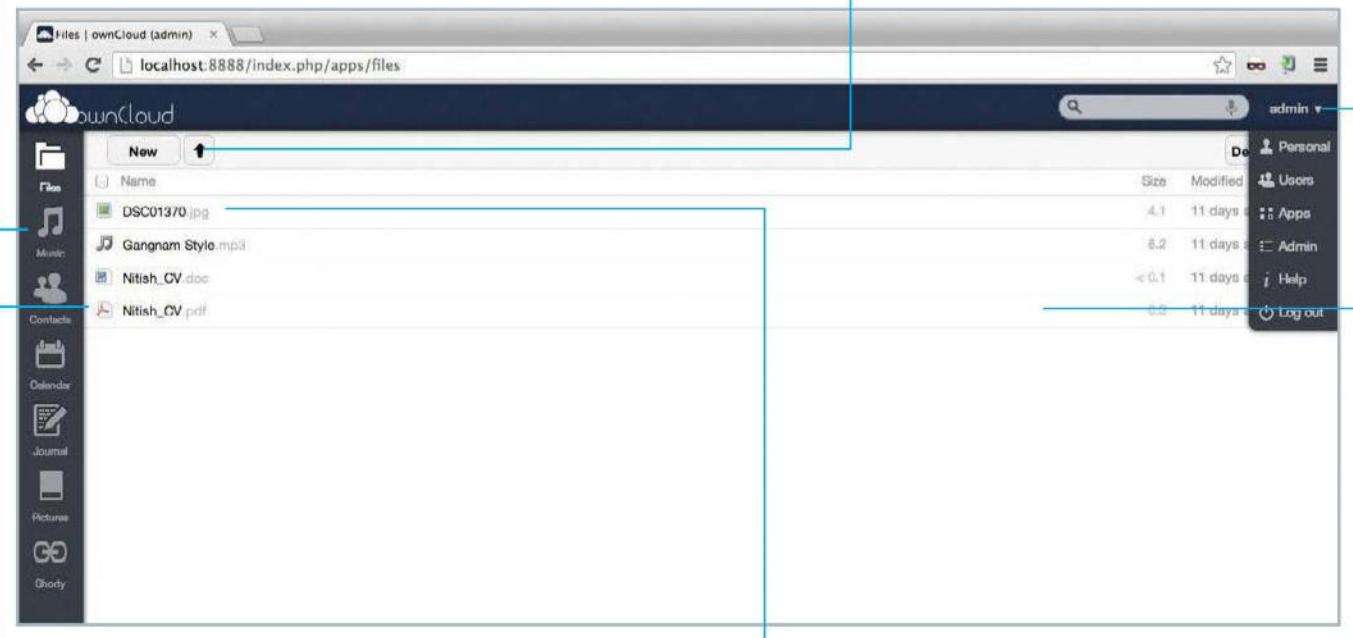
## 09 Export your video

Once you're done with this basic editing, you can create your video so you can play it on other computers or upload to YouTube. Click on Render in the toolbar and select the options you want, including file type, bitrate and resolution you might want to scale the project down to.

ownCloud automatically categorises the uploaded files. Different types can be accessed via the links here

This is the point from where you upload your files to ownCloud

The expanded menu shows the different admin actions available



Handily, there is a built-in PDF viewer available in ownCloud

The built-in image viewer can show the pictures uploaded to ownCloud

Hover the cursor over a file and a menu allows you to rename, download, share or check other versions of that file

# Build your own private cloud with ownCloud

A fast-track guide to setting up your own file management system in the cloud using ownCloud

**ownCloud lets you create your own file management/sharing/backup system without having to rely on a third-party cloud service.** It also provides other functionality like contacts management, calendar management, plug-in support, users, groups etc. All these features make ownCloud a fully fledged enterprise-level file management tool. In this article we will provide a step-by-step installation guide to set up ownCloud on your system, although setting it up on a third-party server (for example a web server provider) will probably be easy because

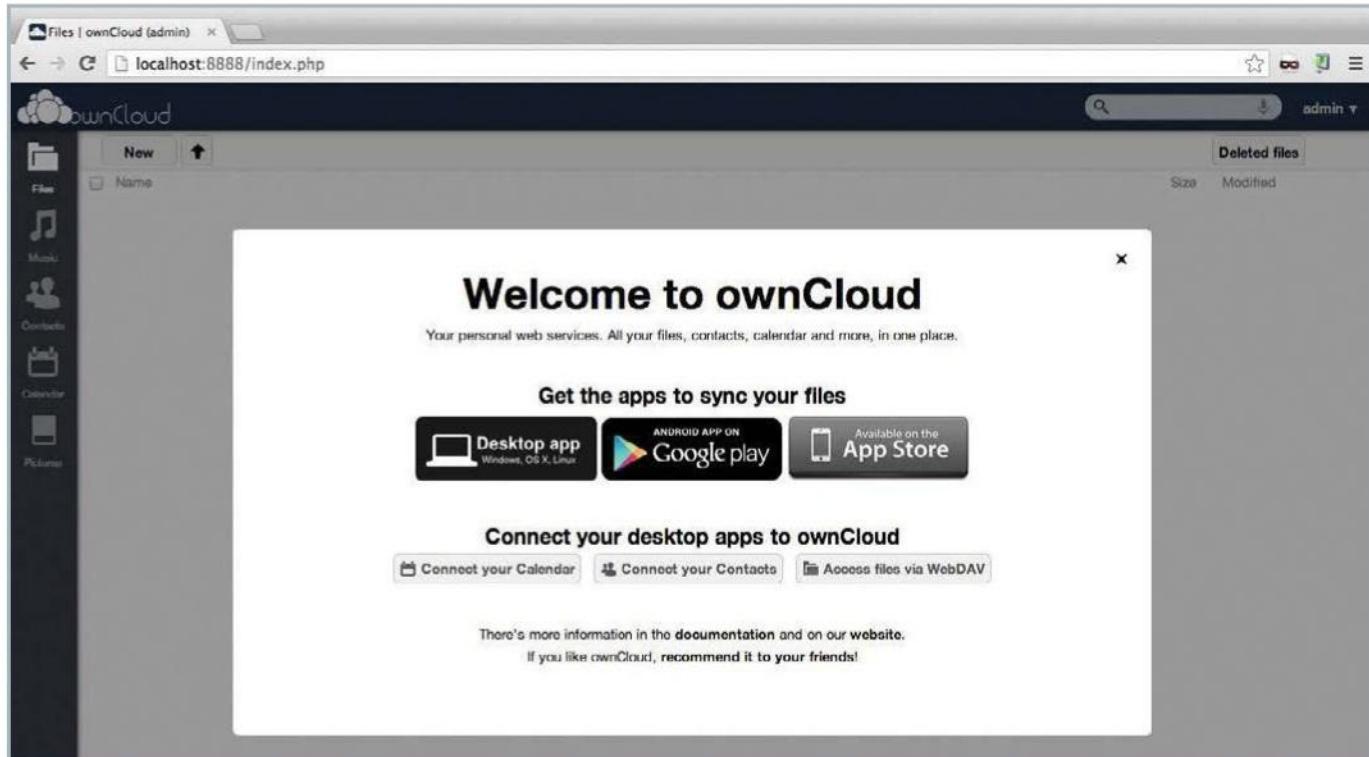
you don't need to install the server. We will then explore some other use cases where it can be deployed. Then we will go through some of the third-party apps/plug-ins available, which can be hooked onto ownCloud to provide further functionality.

We're using the latest ownCloud version 5.0.13 and the dependent PHP versions for this article, although some of the plug-ins discussed here may require older versions of ownCloud. Please check the corresponding plug-in documentation link before trying to install the plug-in.

## Resources

Server like LAMP/WAMP/MAMP

ownCloud plug-ins: [apps.owncloud.com](http://apps.owncloud.com)



## 01 Introduction

ownCloud is an open source file sync and sharing application, available in a free community edition as well as an enterprise edition. It allows you to back up, share and manage files uploaded to the server. With multiple interfaces – like the web UI and Android/iPhone apps – it allows you to be in touch with your data at almost any point of time. The ownCloud desktop client for Windows, Mac OS X and Linux lets you sync your files seamlessly with the ownCloud server, akin to Google Drive or Dropbox clients. With this full ecosystem support for file syncing, ownCloud truly is all about ‘Your Cloud, Your Data, Your Way’, as they say in the documentation.

## 02 Installing the server

There are two possible setups here. You may want to install ownCloud on your system or in a small home/office setup, where it is accessible in a LAN. Otherwise you may want it to be available on the internet. In the latter case, you can skip this step, but if you are planning the former, you will have to identify a machine as the server and install a server such as Apache to that system. But, as discussed earlier, a server is not the only requirement for ownCloud. You also need a database and PHP support. So, a LAMP/WAMP or MAMP server is probably the best way forward,



since all the required tools come bundled with these servers. Your advisor uses the MAMP server in his system to host ownCloud.

## 03 Installing ownCloud

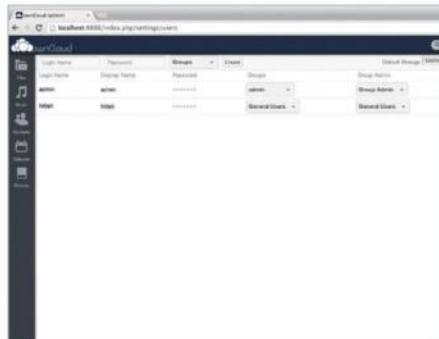
With the server active, we can try to host the ownCloud application. Before that, download the application code from ownCloud’s official website: just click on the ‘Tar or Zip file’ link on the ownCloud install page. Unzip the archive and place the extracted folder in the root of the server. Now, access the folder via your web browser. The welcome page opens up and prompts you to enter the admin ID and password. Below the prompt you’ll also see an ‘Advanced’ link,



where you can change settings related to the database etc. Once done, just click on ‘Finish Setup’ and that’s all! You have your ownCloud.

## 04 Get going with ownCloud

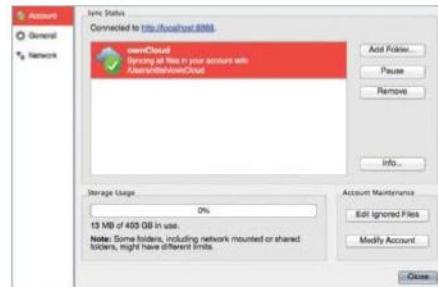
Now that ownCloud is installed, and the admin user created, you may want to add more users and then each of those users may want to upload their files. In this step we will see how to do this. First, the user management – click on the ‘admin’ button in the top-right corner of the homepage. In the drop-down list that appears, click ‘users’. This takes you to the user



management page where you can add/remove users and also segregate them in groups. You also have the option to assign the admin for a group. To upload files, you can simply click on the 'new' button in the ownCloud homepage and then select the file to upload in the upload window.

## 05 ownCloud sync client

Along with the server application, ownCloud also provides the desktop sync client, which can be installed on the user's system. The sync client makes sure any files present in its folder get uploaded automatically to the



server. Just download the client, point it to your ownCloud server URL and enter your credentials. The client then connects to the server seamlessly and starts syncing. The app is highly configurable and lets you change many settings, including bandwidth usage and multiple sync folder support. Also, it is available for all the major platforms: Windows, Linux and Mac OS X.

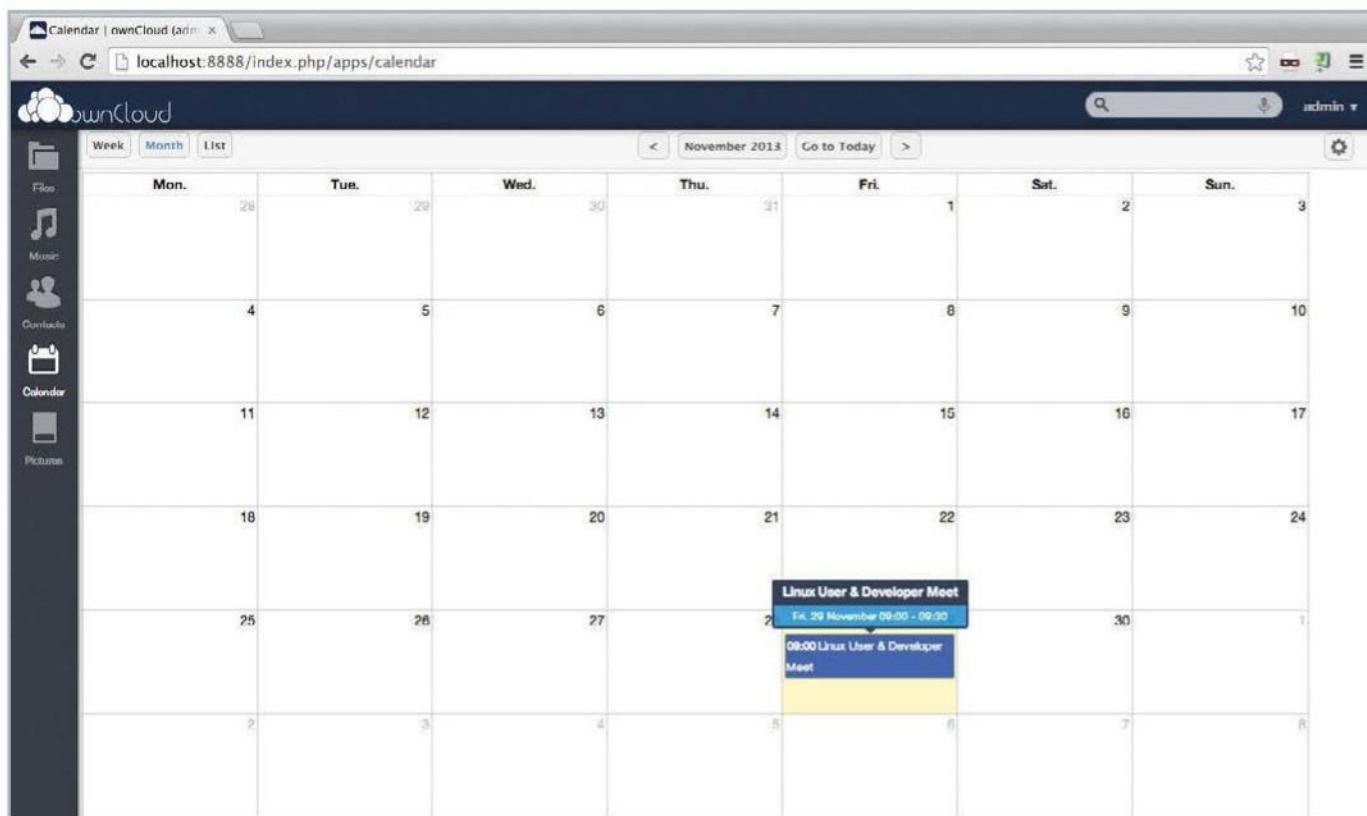
## 06 Other applications of ownCloud

It may appear straightforward, but there is a lot more to ownCloud than just file syncing. ownCloud not only helps sync files between multiple devices and platforms, it also lets you manage your calendar and contacts. The calendar link on the left side

of the homepage lets you create events and share them with other users or groups. It also supports multiple calendars and syncing with iCal. The contacts option supports adding and managing the contact details. Contacts can also be uploaded as .vcf files – and, like files and events, you can even share the contacts among users or groups.

## 07 Roundcube mail plug-in

With contacts and calendar support, you would probably start to think, why isn't email supported? Thankfully, with support for external plug-ins, ownCloud lets you extend the functionality in any way you wish. Open source webmail client provider Roundcube offers an external plug-in which brings your mailbox to ownCloud. To install the plug-in, go to the ownCloud official plug-in portal, download it and paste the downloaded files to the `htdocs/apps` folder. Now, click on the 'apps' link in the dropdown that appears after clicking on admin (at the top-right corner of the page). Here you'll see all the apps; go to the Roundcube app and enable it. This enables the app, but you still need to create the database and configure the web server. For more details, you can check the installation guide.



## “ownCloud not only helps sync files between devices, it also lets you manage your calendar and contacts”



### 08 Shorty plug-in for weblinks management

With the internet having grown so vast, it's become difficult to track webpages and the content you like. On top of that, there are now so many devices: phone, office PC, home PC etc. So, the link you saw at home yesterday is difficult to find when you want to show it to your colleagues in the office. Shorty comes to your rescue. This plug-in lets you create and store short links from the web. It comes as a

preloaded plug-in with ownCloud. To start using it, you just need to enable it from the apps page. Once enabled, you can just drag and drop the button from the Shorty interface to add a site to your list. You can also shorten the URLs with a configurable back-end service like goo.gl, ti.ny etc.

### 09 File encryption plug-in

Security and privacy have become major areas of concern in recent years. With ownCloud you can be sure of using your own server; still, encrypting the data makes the whole setup even more secure. ownCloud ships with an encryption plug-in and once you enable it, all your files get automatically encrypted. The encryption is done server-side and only the ownCloud portal can decrypt the data, using a key that is generated

with your password. So be careful once encryption is enabled because if you forget your password, there is no way to retrieve the data by default. To protect yourself against such loss, you can enable ‘recovery key’ in the ownCloud admin settings for encryption.



### 10 Journal plug-in

The last plug-in we will look at in this article is the journal entry one. This plug-in lets you create journal entries in your ownCloud calendar. The plug-in ships with ownCloud, but before you enable it, enable the TAL Page Templates plug-in (also preloaded with ownCloud). The entries are saved as VJOURNAL records in the calendar and can be sorted and filtered by date/time.

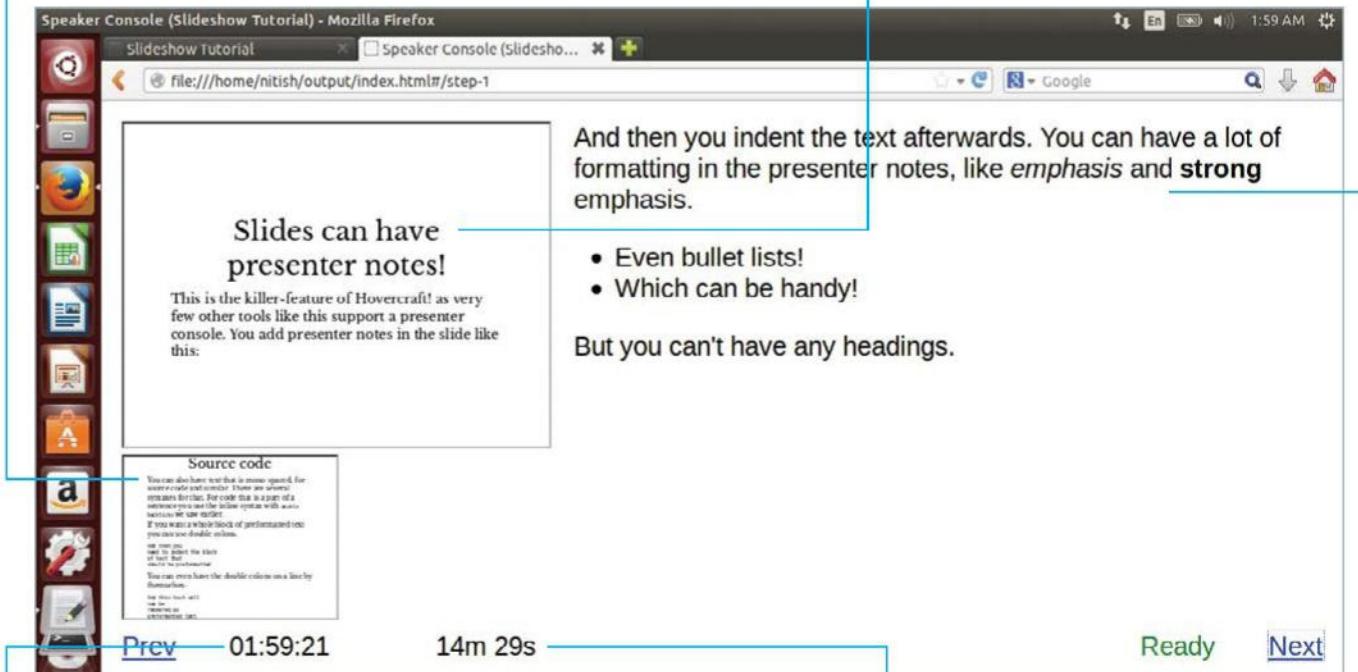
The screenshot shows the ownCloud admin settings page at [localhost:8888/index.php/settings/admin](http://localhost:8888/index.php/settings/admin). The left sidebar includes icons for Files, Music, Contacts, Calendar, Pictures, and Shorty. The main content area has two sections:

- Encryption:** A section for managing encryption passwords and recovery keys. It includes fields for "Recovery account password" (disabled), "Old Recovery account password", "New Recovery account password", and a "Change Password" button.
- Shorty:** A section for configuring the Shorty backend. It shows "Backend: [ none ]" and "Optional configuration of a 'Static Backend'". It includes fields for "Base url:" (Specify a static base url...) and "Example:" (http://localhost/<service>/<shorty\_id>). A note states: "Static, rule-based backend, generates shorty links relative to a given base url. You have to take care that any request to the url configured here is internally mapped to the 'shorty' module. The target of that mapping must be some URL like: http://domain/>/owncloud/>/public.php?service=shorty&relayid=<shorty\_id>"

The next slide is displayed here and prepares you for what's going to be coming up after

This block will show a preview of the current slide you are on, giving you an idea of what the audience is seeing

This is where the slide's notes are shown. They are invisible to viewers but can be seen in console view



The clock shows you the time so you don't need to keep checking to see how long you've been speaking for

The timer shows the time spent on the current slide, which enables you to easily track the time taken without distraction

You can navigate between slides with these buttons. Note that they change the slides in both console view and user view

# Design exciting presentations easily with Hovercraft

Use impress.js and reStructuredText to create presentations using just a notepad

**Presentations are an integral part of our work lives.** Whether you need to present ideas, share knowledge or discuss your company's performance, presentations are your best bet. While a nicely made presentation is a great way to express ideas visually, there can be lots of issues in creating and porting presentations. The current tools that are used to make presentations are all based on GUIs. There are many options, icons and buttons, yet most of them are not being used in 90 per cent of presentations. Because of this, slide editing takes a lot more time than it should.

Another big issue is porting: almost all of us have faced this problem at some point. You add content to your presentation but it is somewhere else when you open it up on another PC.

It is strange that in a field with this many issues, there are few innovations. We will be discussing one innovation, which is a great new way to create and showcase your presentations – Hovercraft. Written in Python, it is based on impress.js and reStructuredText. It lets you create presentations with just a notepad and present them in a web browser.

## Resources

**Python 3.2 or above**  
[www.python.org/download](http://www.python.org/download)

**libxml and libxslt**  
[www.xmlsoft.org/downloads.html](http://www.xmlsoft.org/downloads.html)

**Hovercraft documentation**  
[www.github.com/regebro/hovercraft](http://www.github.com/regebro/hovercraft)



## 01 impress.js

As we said earlier, Hovercraft is based on impress.js and reStructuredText. First let's learn more about impress.js and its capabilities. Impress.js is an open source, JavaScript library, aimed at helping you make better presentations. It uses the power of CSS3 transforms and transitions, which is supported by almost all the latest browsers. Some of the effects that impress.js is capable of are: pan, rotate, zoom, scale and even 3D effects. When it's all finished, the resulting impress.js presentations are basically HTML files. Check out the impress.js demo here: <http://bartaz.github.io/impress.js>.

You may think, what is the need of Hovercraft, if impress.js in itself is a presentation framework, capable of everything? The reason is because impress.js requires you to write HTML and CSS to create presentations – not everyone would like to do that, as it calls for a steep learning curve. Hovercraft combines the power of impress.js with the ease of reStructuredText to give you a great new way to create presentations easily.

## 02 reStructuredText

Another very important component of Hovercraft is the reStructuredText. It is with the help of reStructuredText that you can write

plain text files, which then get interpreted into HTML. This output then serves as the input for impress.js. For the uninitiated, reStructuredText is a WYSIWYG plain text markup syntax and parser system. In simple terms it lets you write text, based on defined syntax, in any text editor and then parses it to HTML (there are steps in between, but skip them for now). It is widely used as an inline code documentation tool or even a standalone document generator. ReStructuredText aims to be readable and unobtrusive so that it can be easily understood, which makes it easy for you to create more standardised documents. To give you an idea about how reStructuredText works, let's take a look at a few of the markup rules:

Plain Text	Typical Result
*Emphasis*	<b>Emphasis</b>
**Emphasis**	<b>Emphasis</b>
``Inline literal``	Inline literal
Title =====	<b>Title</b>
Subtitle -----	<b>Subtitle</b>

This is the quick reference guide: [docutils.sourceforge.net/docs/user/rst/quickref.html](http://sourceforge.net/docs/user/rst/quickref.html).



## 03 Installation

Now that you are aware of the various components, let's look at installing Hovercraft. Hovercraft is written in Python and it needs Python 3.2 or above to run properly. You also need pip – a package manager for Python that lets you easily install and manage packages. Note that, with the latest version of Python 3.4, there is no need to separately install pip as it comes included. If you are on Windows or Mac, you also need the libxml and libxslt for the Hovercraft installation. Download them from [bit.ly/1tEs3Ra](http://bit.ly/1tEs3Ra).

In this tutorial we have used Ubuntu 14.04 as the host OS, which is preloaded with Python 3.4. So if you are on Ubuntu, you just need to type `sudo pip3 install hovercraft` in your terminal as all the dependencies are already present. If everything goes fine you will get the message "Successfully installed Hovercraft".



## 04 Create your first presentation

To create your first Hovercraft presentation, fire up Gedit/Leafpad/Vim or any other text editor of your choice and start writing your content. Just keep in mind that the text needs to be written in reStructuredText syntax. For example, to define the slide header text you will need to underline the text so that it is parsed as a heading by Hovercraft. Note that when you underline two different lines, the position of the lines defines the title and the subtitle. To create bullet point or numbered lists you just need to indent to the right and write the points as you would in a word document. There's more info on styling your presentations in the later steps. After you are done with the text file, save it with the '.rst' extension. We saved our file as sample.rst. You can then run the command `hovercraft <presentation> <targetdir>` in your command prompt; replace `<presentation>` with your rst file name and `<targetdir>` with the folder you'd like to be created with the output. You can then open the index.html file that gets created in the output folder using your browser. That's all there is to it! You now have your first Hovercraft presentation.

## 05 Styling presentations – Hovercraft rules

We've already mentioned several of the reStructuredText rules; there are more which are well-documented in the quick reference guide we mentioned in Step 02. It's worth being aware that in a few cases Hovercraft uses the rules in specific ways:

- The transition in reStructuredText is used to separate slides. A transition is simply a line with four or more dashes, ie -----. So you need to add a transition to indicate the end of a slide and the beginning of another.
- Hovercraft makes the first slide only after it finds a transition. Anything before it belongs to the presentation.
- Any impress.js field added in the presentation before the first slide is

```

sample.rst x
:title: Sample Presentation
:author: Nitish Tiwari
:description: Sample presentation for demo purposes.
:keywords: presentation, restructuredtext, impress.js, tutorial
:css: sample.css

This slide show is a sort of tutorial of how to use Hovercraft! to make
presentations. It will show the most important features of Hovercraft! with
explanations.

-----
The first slide
=====

Any line underlined with punctuation is made into a heading. Each type of
underlining will be made into a different level of heading. The order of
headings encountered, decide the level of the heading. Next slide will give
you a better picture.

-----
First header
=====

You can choose other punctuation characters as your level 1 heading if you like,
but this is the most common. Any of these characters work::

= - ^ : ' " ~ ^ _ * + # < > .

```

## 04 Create your first presentation

# “Hovercraft converts the reStructuredText into XML then uses XSLT to translate that into HTML”

rendered into the attributes of the main impress.js `<div>` tag. Currently it is used to set the transition duration with `:data-transition-duration: field`.

- Similarly any impress.js field added into the first slide is rendered into attributes on the slide `<div>` tag. This is used to add special effects to the slide, as well will explain momentarily in Step 07.

## 06 Styling presentations – adding external files

Adding external images or style sheets to your presentation is extremely easy with Hovercraft. To add images to a slide just write:

```
.. image:: path/to/image.png
:height: 600px
:width: 800px
```

Similarly, you can add external CSS and there are different ways to do that. The first way is via templates – whenever you use a template, the corresponding CSS automatically gets included. For example, if you use the default

template, three CSS files are added to your presentation: highlight.css, hovercraft.css and impressConsole.css. Another way is using the `:css:` at the top of presentation:

```
:css: css/presentation.css
```

The third way involves adding CSS with the command line parameter:

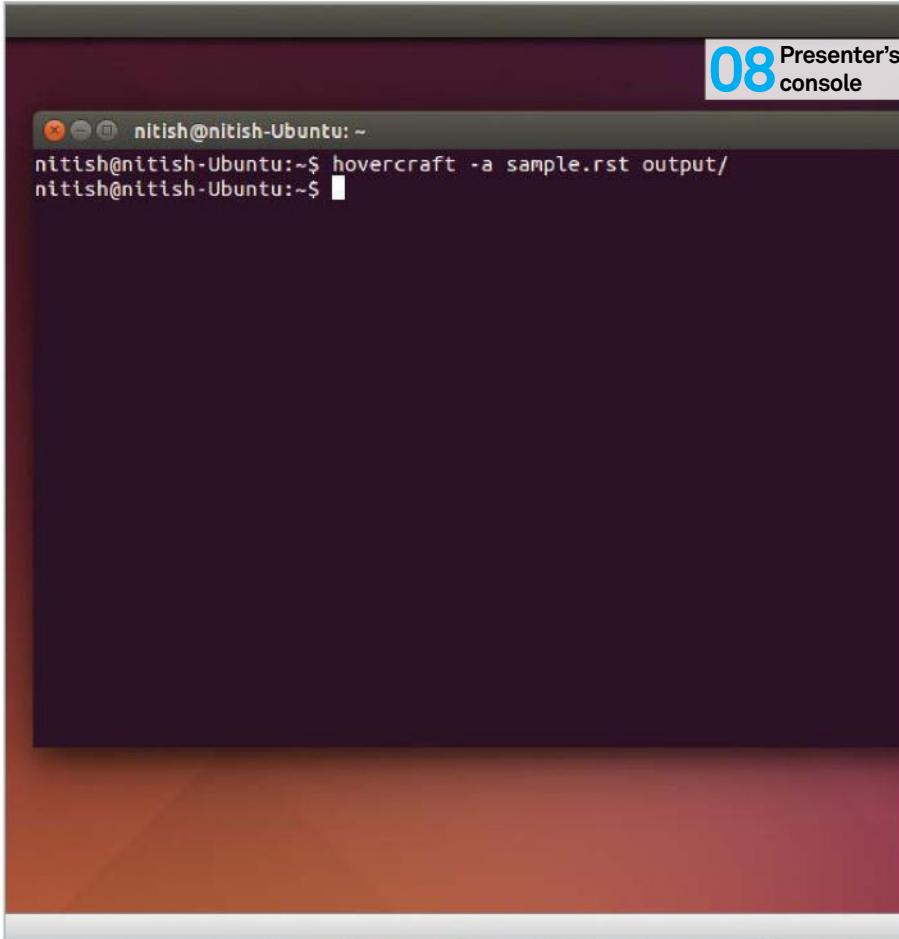
```
hovercraft extra-css=my_extra.css
sample.rst outdir/
```

You can also style a single slide with the help of the `:id:` field. Add the field into the slide content so that it looks like this:

```
:id: slide-id
```

Then add the custom code like this (to any CSS already included in the presentation):

```
div#the-slide-id {
    /* Custom CSS here */
}
```



## 07 Special effects

Hovercraft lets you create some awesome text effects like pan, rotate, zoom and even 3D and this is done using impress.js fields. You can add these fields before a slide header and then the slide will have the effect automatically applied. Now let's look at various effects and corresponding fields that are available to us:

- A pan effect can be achieved using the :data-y: field of impress.js, which is used to indicate the vertical position of a slide in pixels. It can be negative as well
- To rotate a slide you can use the :data-rotate: field and then enter the angle of rotation
- Zoom effect can be achieved using the :data-scale: field. It defaults to 1, and a value greater than 1 indicates zooming out while values less than 1 indicate zooming in
- 3D effects can be achieved with impress.js using the :data-perspective: or the :data-rotate-x: field.

## 08 Presenter's console

This is another awesome feature of Hovercraft. It lets you see the current and next slides along with the notes you added for each slide. It also shows a clock and a timer to help you rehearse and get an idea of time taken to present each slide. Notes can be added to slides like this:

.. note::

Add your note here!

Add this after the slide's content; otherwise it will create blank space in the middle of your slide's content. To view the presenter's console, just add the -a flag while creating the presentations. Now when you open the presentation via a browser, the presenter's console should pop up to a new tab, you'll need to disable your pop-up blocker though. Since the presenter's console appears in a separate tab, you can keep it running during live presentations as well – just remember to share the correct tab to your viewers.

## 09 Templates

Templates are a way for you to predefine the look and feel of your Hovercraft presentation. By default, Hovercraft has two templates available to choose from – the first is the default template, which is to be used if you don't provide the template name during the creation of your presentations, and the second is simple: a template without the presenter's console feature included. If you're not satisfied with any of these options, you can add one yourself. Note that Hovercraft generates presentations by converting the reStructuredText into XML and then using XSLT to translate the XML into HTML. So, a new template needs a template XSL file, a configuration file and any number of CSS/JS or other resource files.

Configuration files define the content of the template. It has only one section, [hovercraft], with subsections defining the template content. As good practice you should name it as 'template.cfg', but you can use any file name with .cfg as the extension. Remember to pass the name of the file (while using the Hovercraft command) if you are not using the default name.

With template files, the file specified in the template parameter of configuration file is the actual XSLT template that performs the XML to HTML transformation. In most cases you can just take the file template.xsl from the templates folder in the Hovercraft folder and edit it yourself.

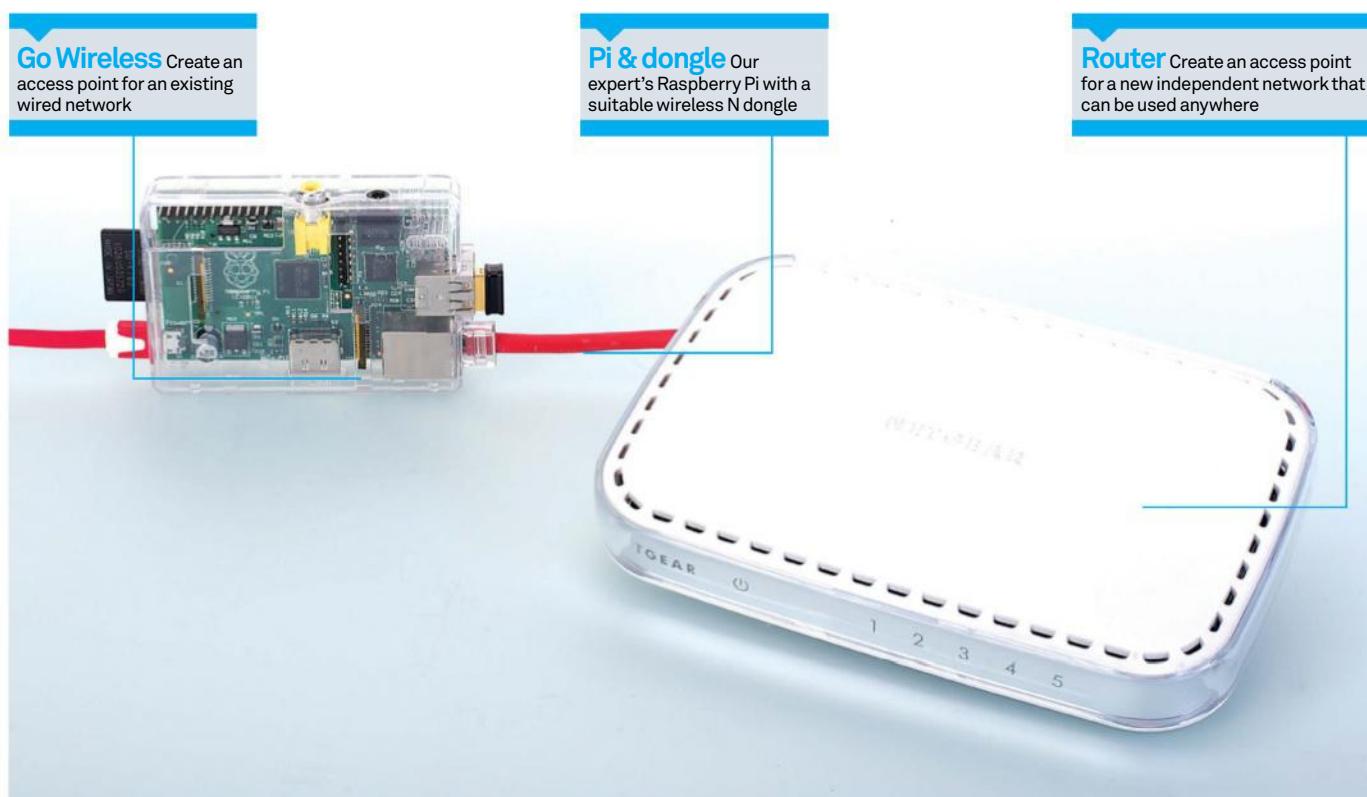
## 10 Portability

It is very common for a presentation to be made on one computer and then presented on another. In such scenarios it is not guaranteed that both the computers have even the same operating system, let alone the same set of tools installed onto each of them. Hovercraft covers you well here; it is compatible with all the major browsers. One important thing to note here is that different browsers may render fonts differently. The best solution to this problem is to include fonts with the presentations – even better if you download them before including them; this way you can avoid unnecessary surprises, and your presentation will be truly platform independent!

To include a font, you can download it and save it in a folder called fonts inside the folder storing your presentation. Later, just edit the CSS @font-face rule. If you plan on using web fonts, grab them with @import statement. Note that if you are using default template, you need to edit the /hovercraft/templates/default/css/hovercraft.css file. If you are using any other template, edit the correspondong .css file.

# Set up a wireless access point with a Raspberry Pi

How to wirelessly connect to your Raspberry Pi, or any existing network connected to it



## What you'll need

- Latest Raspbian Image  
[raspberrypi.org/downloads](http://raspberrypi.org/downloads)
- A router or switch
- A USB wireless dongle that supports Access Point mode
- A Wi-Fi device to test with

A Raspberry Pi with a wireless dongle that supports Access Point mode is a very versatile tool. As well as connecting to existing wireless networks, the Raspberry Pi can become an access point and have multiple devices connected to it. It can either have its own network or bridge onto an existing one and make it wireless. One potential application of this is to create a wireless access point for guests, with appropriate firewalling rules to make sure they can't connect to anything on your LAN. Another is simply connecting to it over wireless rather than Ethernet, which can be useful if the Ethernet port is already in use.

### 01 Install the required software

Log into the Raspbian system with the username **pi** and the password **raspberry**. Get the latest package lists using the command:

```
sudo apt-get update
```

We'll be using **hostapd**, which is a daemon that handles access point management and authentication. We'll also need **bridge utils**, which will be used to bridge the Ethernet and wireless interfaces together. **iw** is used to get information about the wireless interface, and **dnsmasq** is used as a DHCP server.

Install these with:

```
sudo apt-get install hostapd bridge-utils iw dnsmasq
```

## 02 Check for AP mode

It's crucial that you have a wireless dongle that supports Access Point (AP) mode. To verify that, connect the wireless dongle and type `iw list`. There will be a section titled 'Supported interface modes'. Ensure that AP mode is in that list. Our expert's output looked like this:

Supported interface modes:

- \* IBSS
- \* managed
- \* AP
- \* AP/VLAN
- \* WDS
- \* monitor
- \* mesh point

## 03 Bridge interfaces

We'll bridge the Ethernet interface and the wireless interface so they can share the same IP address and traffic can pass between them. To do this, edit `/etc/network/interfaces` (using sudo) and change it to look as follows:

```
auto lo br0

iface lo inet loopback

allow-hotplug eth0
iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet manual

iface br0 inet dhcp
    bridge_ports eth0 wlan0
    bridge_waitport 0
```

At the moment, we'll be bridging onto an existing network, but we will set up our own later on. Reboot using `sudo reboot` so that the changes take effect.

## 04 Configure hostapd

We first need to edit `/etc/default/hostapd` to start the daemon on boot. Uncomment the 'DAEMON\_CONF' option and replace it with:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

We then need to create the hostapd config file at the path specified above. It should have the following contents:

```
interface=wlan0
bridge=br0
driver=nl80211
country_code=UK
ssid=pinet
hw_mode=g
channel=1
wpa=2
wpa_passphrase=super_secret
wpa_key_mgmt=WPA-PSK
ieee80211n=1
wmm_enabled=1
```

```
bridge_waitport 0
address 192.168.0.1
network 192.168.0.0
netmask 255.255.255.0
```

## 07 Configure a DHCP server

We installed dnsmasq in the first step. Configuring it is really simple. Start by taking a copy of the default config:

```
sudo cp /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

Then edit `/etc/dnsmasq.conf` to contain the following lines:

```
interface=br0
dhcp-range=192.168.0.2,192.168.0.254,255.255.0.12h
```

Reboot for the changes to take effect. You'll want to disconnect the Raspberry Pi from your existing network at this point, otherwise there will be two DHCP servers on the network.

## 08 Try it out

Devices on the wireless interface will be able to talk to devices on the wired interface, and any device plugged into the Pi will get an IP address so that it can talk to the other devices.

## Skills to learn

**1 Connect** It's a very useful and convenient form of device connectivity and used by many add-on peripherals.

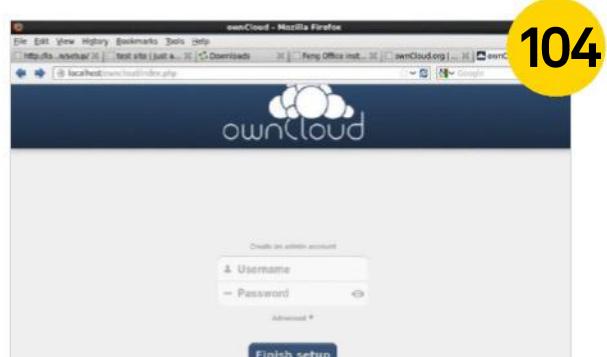
**2 Networking** The world of networking is something of a dark art. Projects like this make it much less like magic.

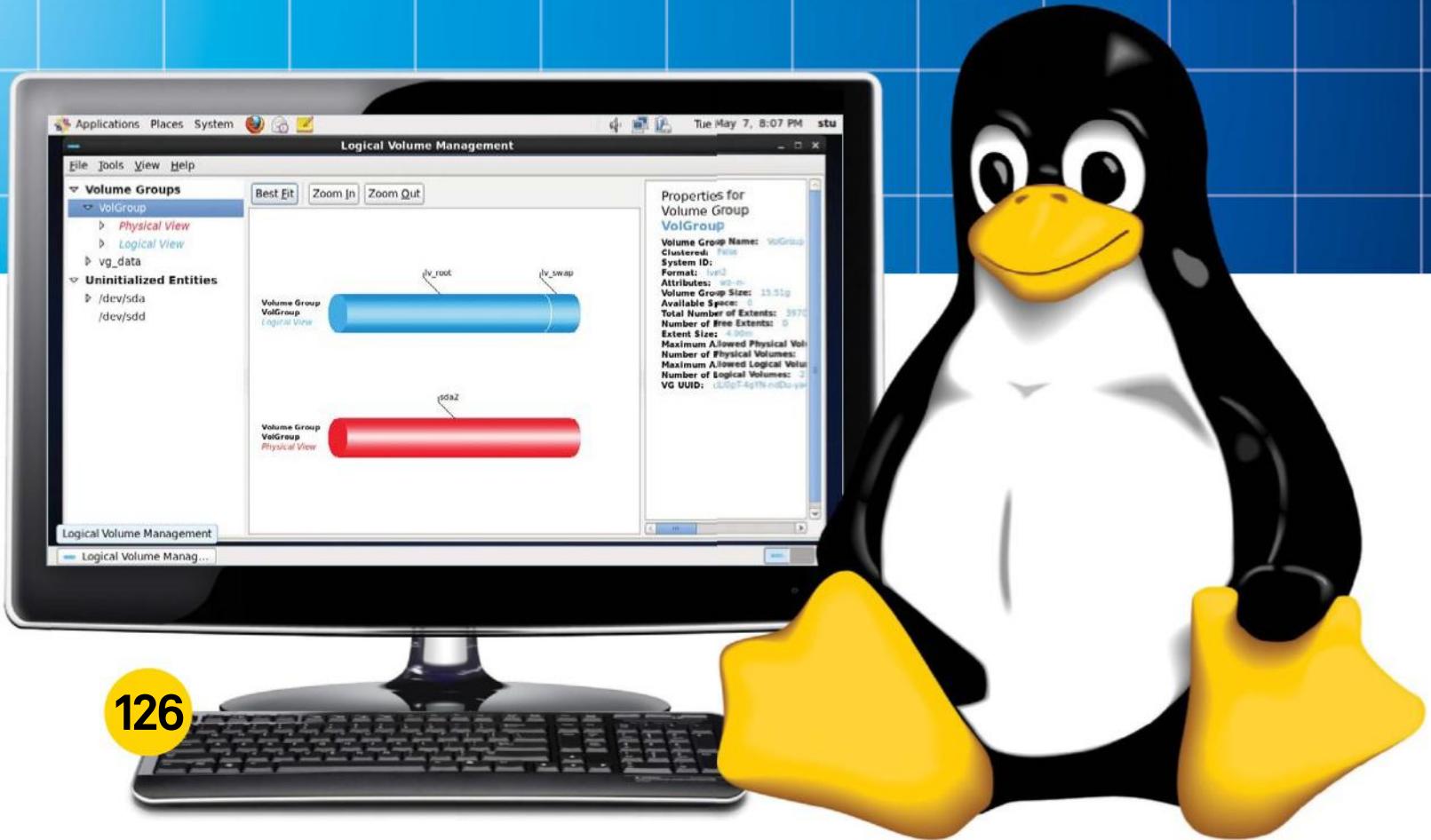
**"As well as connecting to existing wireless networks, the Pi can become an access point"**

# Hacks

## Customise and tweak your system

- 104 Turbocharge your cloud**  
Use the power and efficiency of a lightweight solution
- 110 Speed up Linux with Openbox**  
Speed up your day-to-day computing without sacrificing usability
- 114 Bypass restrictive firewalls using SSH tunnelling**  
Create secure network connections on the fly
- 118 Create a custom build of Gentoo**  
Build a custom distro from the ground up





126

**“We’ll take you through the steps you need to compile your own customised kernel”**

### 122 Create a custom Linux kernel to optimise performance

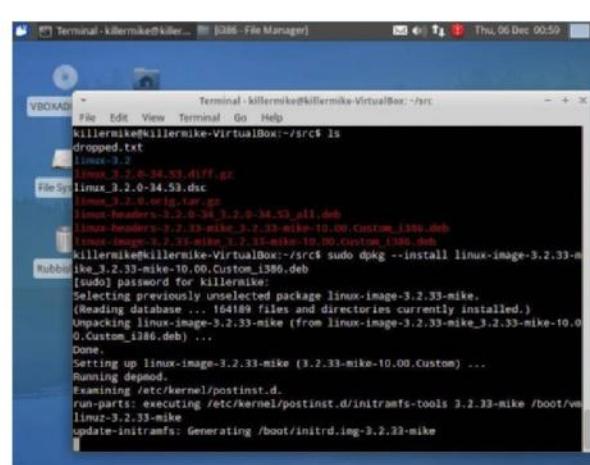
Compile your own customised kernel for performance and specialised use

### 126 Resize your disks on the fly with LVM

Never reformat and restore your drive again

### 130 Scrape Wikipedia with Beautiful Soup

Parse Wikipedia’s HTML and store it for offline reading



122

# Turbocharge your **CLOUD**

Build your own cloud services using the power  
and efficiency of a lightweight solution



The value of an efficient web server comes into its own when serving cloud applications over the web or an internal network. This is because every increase in efficiency increases the total number of clients to which you can simultaneously provide services. For this reason, this project employs Nginx, a lightweight, performance-optimised web server, rather than the more common Apache. In most cases, Nginx requires extra configuration to support these services, and we'll cover how you go about doing that. In this example, we're going to use a fresh installation of CentOS 6 as the

host operating system. This tutorial makes use of CentOS 6.4, but the instructions should be much the same for Red Hat and Fedora.

Once we have the Nginx server fully working, we shall add some typical cloud applications. WordPress is a blogging application and relatively easy to install. Feng Office Community Edition is an open source business application that offers word processing alongside team-orientated facilities such as planning and time tracking. OwnCloud is a file sharing application at heart, but offers some other nice facilities such as a media browser.

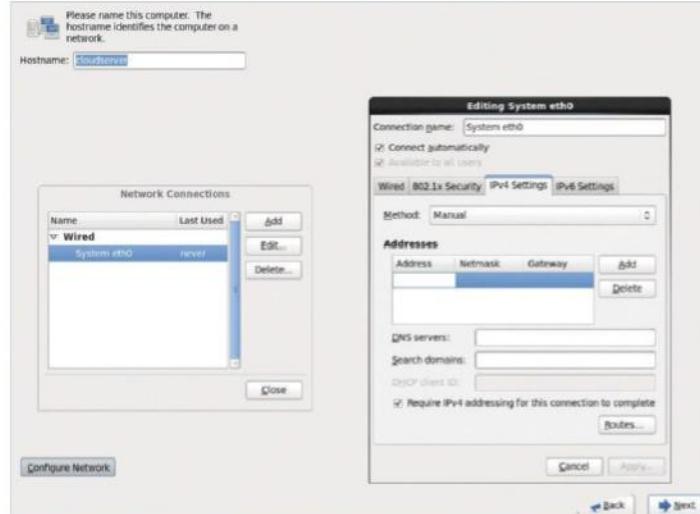
## Set up the server OS

Begin by installing CentOS. While installing, you can accept most of the defaults, but it's a good idea to customise the network setup during installation. Firstly, give the computer a meaningful hostname such as 'cloudserver'. On the same page of the installation, select the Configure Network button in order to bring up the Network Connections dialog. Select the network adaptor that you are going to use to connect to clients on your LAN or the internet and then click on 'Edit...'. The first thing to select here is the 'Connect automatically' option. It's typical to set a static address in the case of a server rather than allowing DHCP to assign one automatically. To do this, select the IPv4 tab and change the 'Method' drop-down from DHCP to Manual. Now click on Add and add an IP address. This should be congruent with the address layout of the rest of your network. So, for example, if your machines have an IP address that begins 192.168.0.x, an IP address of 192.168.0.100 with a netmask of 255.255.255.0 and a gateway 192.168.0.1 should be suitable. Typically, the DNS server address provided by your ISP should be entered. Click Apply when finished.

We recommend selecting the 'desktop' installation profile, presuming you want a system with a GUI. After rebooting, log in, open a terminal and type `su` to become root. You will have to carry out nearly all of the operations in this tutorial as root.

### Install server components

We'll add the the Nginx repository to the system next. To manually add the source repository, create a file called `/etc/yum.repos.d/nginx.repo`



■ Setting the network options for static IP while installing CentOS



and add the following:

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/
centos/6/$basearch/
gpgcheck=0
enabled=1
```

We need to configure the way that Nginx accesses PHP-FPM. Add these lines:

```
location ~ \.php$ {
```

■ CentOS installed and ready for reboot

### TIP How to pronounce Nginx:

If it ever comes up in a real-life conversation, say it like 'engine ex'.

```
root /usr/share/nginx/html;
try_files $uri =404;
fastcgi_pass 127.0.0.1:9000;
fastcgi_index index.php;
fastcgi_param SCRIPT_FILENAME
$document_root$fastcgi_script_name;
include fastcgi_params; }
```

### Make the services permanent

Ensure that the MySQL server (named mysqld), PHP-FPM (named php-fpm) and Nginx (nginx) services start without errors by typing `service [service name] start`. Once this has been checked, type `chkconfig --add [service name]` to cause them to start on boot. Type `service --status-all` to list all running services. This list is quite long, and you can filter it using grep like this: `service --status-all | grep sql`. You can find more detailed information about a running service by typing `service [service name] status`.

```
killermike@centserver:/usr/share/nginx/html$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.1.66 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

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

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

mysql> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.01 sec)

mysql> GRANT ALL PRIVILEGES ON wordpress.* TO "wordpressusername"@"hostname"
-> IDENTIFIED BY "zack";
ERROR 1470 (HY000): String 'wordpressusername' is too long for user name (should be no longer than 16)
mysql> GRANT ALL PRIVILEGES ON wordpress.* TO "wordpressuser"@"hostname" IDENTIFIED BY "wordpresspass";
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> exit
Bye
[root@centserver html]#
```

■ Creating a MySQL database. A handy skill to have when working with a web server

## Add webmail

If you are offering services via the cloud, you might already be using webmail such as Gmail. However, if you have tight requirements for how people use email, you could consider adding your own front-end. This usually takes two possible forms. You can add your own IMAP server such as Cyrus ([cyrusimap.web.cmu.edu](http://cyrusimap.web.cmu.edu)) or Dovecot ([www.dovecot.org](http://www.dovecot.org)), both available from the CentOS package repository. You then install a IMAP web front-end into Nginx using the techniques that we've already covered, alongside the specific documentation for those packages. SquirrelMail ([squirrelmail.org](http://squirrelmail.org)) is an lightweight front-end and Roundcube ([roundcube.net](http://roundcube.net)) is more fully featured.

The second approach is to host an IMAP front-end, but then use a webmail service such as Gmail as the IMAP back-end rather than running the server yourself. This allows you to customise the experience along with all the benefits of off-site email.



■ Logging into SquirrelMail

## TIP Fetching the latest WordPress

The latest WordPress version is always available from a fixed URL ([wordpress.org/latest.tar.gz](http://wordpress.org/latest.tar.gz)). We wish more open source projects would do this!

## Install cloud applications

### WordPress

Let's provide our users with blogging facilities using WordPress. This is good first test for our server because WordPress is extremely well documented and relatively easy to install. If this won't install, something has gone seriously wrong with the setup, and yet it's extensive enough to fully test the server. Go to the WordPress site and download the latest version. If you prefer, you can always directly obtain the latest version of WordPress by typing `wget http://wordpress.org/latest.tar.gz`. Once you have it, type `tar -xzvf [name of archive] -C [output directory]` to unpack it. We can find the root directory of the web server by locating the line 'root' inside default.conf. By default it should be '/usr/share/nginx/html'; and this is the directory in which we need to unpack the archive. Before we can begin the installation, we need to set up the MySQL database that it needs.

### Create MySQL database

Experience with MySQL is one of the most useful skills to have when working with web servers. Creating a MySQL database and user is a procedure that we will be repeating during this tutorial. Typically, each cloud application should use its own MySQL database with its own user. We use the Root user account that you set up earlier on to administer all of these users and databases. Note that this 'Root' user has no relationship to the privileged user account that you use to administer Linux systems;

## TIP Downloading CentOS

CentOS is now provided as two DVD images. It might be a good idea to give preference to using a torrent, which is usually the fastest source and located next to the ISOs in the repository.

they just happen to have the same name. Restart the MySQL server by typing `service mysqld restart`. We're going to use 'mysql', the command-line tool, to create the MySQL database. The syntax of MySQL queries look quite intimidating, but they're actually fairly easy to work with. Some tutorials create the user of each database with a separate command but it's not really needed for this work, and we will just use the GRANT command. We create a new database like so:

```
CREATE DATABASE [name of database];
```

We set up a new user like so:

```
GRANT ALL PRIVILEGES ON [database name.*] TO "[name of user]"@[domain (usually localhost)]" -> IDENTIFIED BY "[password]".
```

As mentioned earlier, this creates a new user as well as assigning a password to it. Now let's get to work setting up a database and user for WordPress.

Log into the MySQL client by typing `mysql -u adminusername -p`. You will be prompted to enter your password and when you have, you should find yourself on the 'mysql>' prompt. Type `CREATE DATABASE wordpress;` to create the database. This should output:

```
Query OK, 1 row affected (0.01 sec)
```

■ Adding the MySQL database settings during WordPress installation

■ Using the WordPress Admin panel, having logged in for the first time



■ The Feng Office word processor is highly integrated with the other tools

■ Entering the database details in the Feng Office installer

```
default.conf wp-config.php
/*
 * @package WordPress
 */

/** MySQL settings - You can get this info from your web host */
/** The name of the database for WordPress */
define('DB_NAME', 'database_name_here');

/** MySQL database username */
define('DB_USER', 'username_here');

/** MySQL database password */
define('DB_PASSWORD', 'password_here');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 * You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in again.
 *
 * @since 2.6.0
 */

```

■ Manually adding the Nginx repository using nano as the text editor

Type GRANT ALL PRIVILEGES ON wordpress.\* TO "wordpressuser"@"localhost" IDENTIFIED BY "password"; substituting the password for one of your own choosing. This creates the user and assigns a password. Note the semicolons that end each command. The response should be:

Query OK, 0 rows affected (0.00 sec)

Now type in FLUSH PRIVILEGES; followed by quit;.

## Complete installation

We'll now edit the WordPress configuration and then run the script that completes the installation. Make sure that you are in the /usr/share/nginx/html directory before starting. Create a copy of the example configuration file by typing cp wordpress/wp-config-sample.php wordpress/wp-config.php. Now

open the file in a text editor. Set the fields 'DB\_NAME', 'DB\_USER' and 'DB\_PASSWORD' so that they equal, respectively, the name of the MySQL database that you created (wordpress), the name of the WordPress user that you created in the database (wordpressuser) and the password that you created for that user. Save the file.

Now navigate your web browser to localhost/wordpress/wp-admin/install.php. If everything has gone okay, you should now receive a message asking you to create a default site. Call it anything you like, as we can change it later.

### TIP Make a symbolic link

If you prefer, you can make a symbolic link to the web server directory to save on typing. For example, ln -s /usr/share/nginx/html /webservice (note the space) allows you to just type '/webservice'.

### TIP Note all passwords

We'd advise you to create an empty document called 'my passwords' at the beginning of this project, as you're going to have to record quite a lot of usernames, passwords and database names.

## Create an entry page

Create an entry page for your services. Here's an example. Replace 'localhost' with the domain name or IP address of your server when you're ready to take it out of testing and make it accessible to other machines. Save this as index.html and then place it in /usr/share/nginx/html/, the web server root.

```
<!DOCTYPE html>
<html>
<body>
<h1>Welcome to cloud services</h1>
<h2>Please select service</h2>
<a href="http://localhost/owncloud/">Owncloud - Filesharing and remote storage</a><br>
<a href="http://localhost/feng/">Feng Office - Office and collaboration features</a><br>
<a href="http://localhost/wordpress/">Wordpress - An easy to use blogging system</a><br>
</body>
</html>
```

Enter an admin password and your email address and then click on Install WordPress. Log into the site with your username and password as a final check that everything is working okay.

### Feng.net Community Edition

Feng Office offers an impressive suite of office facilities, including word processing, time management, contact management, calendar and presentation creation among many other features. It's all tied together in an integrated suite that encourages team collaboration in a business context. We'll be working with the free, open source Community Edition.

```
Begin by installing its dependencies: yum
install php-ldap php-mysql php-gd
php-imap php-odbc php-pear php-xml
php-xmlrpc. Restart Nginx and PHP-PFM:
#service nginx restart
#service php-fpm restart
```

Pay a visit to the Feng Office website ([www.fengoffice.com](http://www.fengoffice.com)) and download the Community Edition, supplied as a zip file. Decompress it in the web server directory with:

```
unzip fengoffice_2.3.zip -d /usr/
share/nginx/html/feng/. Now cd to that
directory and type chmod 777 config tmp
upload cache to give full access to those
directories. We then create a MySQL database
for Feng to use, just like we did for WordPress.
Type mysql -uroot -p to begin.
```

```
mysql> create database fengoffice;
mysql> GRANT ALL ON fengoffice.* TO
```

#### **TIP** Gedit

To launch a GUI text editor such as gedit while root under CentOS, use sudo gedit [filename] & This sorts out the environment variables for you and detaches the terminal.

### Set up port forwarding

In order to enable machines outside of your LAN to connect to your server, you need to enable port forwarding. How you go about this varies depending on your make and model of router. If unsure of how to do it, visiting [portforward.com](http://portforward.com) is a good start because it has router-specific instructions. At the same time, use the router setup pages to discover your external IP address. Currently, searching for 'IP address' in Google will also display your current external IP address.

You can discover the IP address of your server by typing 'ifconfig' into it. You then follow the instructions specific to your router (usually accessible by using the browser to visit the first IP address on your network such as 192.168.0.1 or 192.168.1.1). You must tell it to forward ports 80 and 443 to the IP address of your server. Then, you should be able

to access the server from outside of your network by pointing a browser at the external IP address of your router.

To save having to remember this IP address, you could buy a domain name and point it at your network. However, if your ISP does not offer static

IP for customers, a dynamic DNS server might be a good alternative. Organisations such as No-IP ([www.noip.com](http://www.noip.com)) and ([www.opendns.com](http://www.opendns.com)) offer a free subdomain that can be updated via a web interface each time you disconnect from the internet and reconnect.



■ Logging into the router setup page

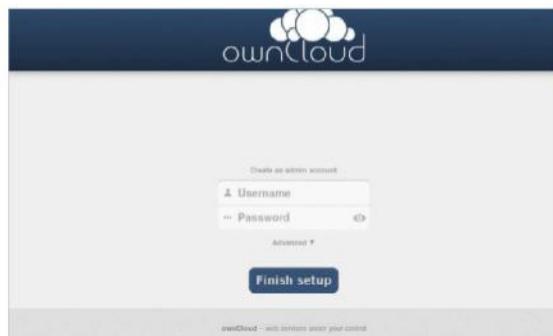
```
fenguser@localhost IDENTIFIED BY
'your password';
mysql> flush privileges;
mysql> quit;
```

In a web browser, go to `localhost/feng/`. This should bring up Step 1 of the installation sequence. Clicking Next takes you to Step 2, which carries out a check of the server environment. Presuming that everything looks okay, click Next and proceed to Step 3. Here, you must enter the username, password and database name of the MySQL database that you just created. Clicking Next should allow you to log into your new Feng system.

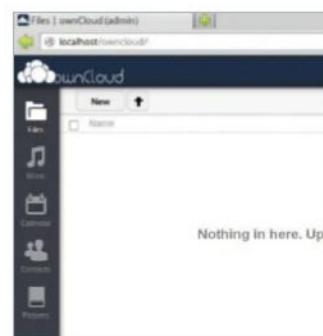
**TIP** **Use tab completion**  
Don't be intimidated by those long
pathnames and filenames, use tab completion.
It's faster and often guards against mistakes.

### OwnCloud

OwnCloud is primarily a service for personalised file sharing. The idea is that you can access your files from anywhere and from any device, while also having the option of selectively sharing files with other people. Download the current version of ownCloud from [owncloud.org](http://owncloud.org). This normally comes supplied as a .tar.bz file, so the switches to decompress are slightly different. Type `tar -jxf [archive name] -D /usr/share/`



■ Logging into ownCloud for the first time



■ The ownCloud file browser

### What is PHP-FPM?

PHP-FPM is a FastCGI implementation. CGI is the method that web servers use to call an executable on the host. These executables are typically written in a scripting language that the host understands, such as Perl or PHP. Traditionally, web servers opened each executable in its own process, an approach which aided simplicity. The problem is that opening and then closing processes with each request made upon the server doesn't scale very well. For this reason, FastCGI servers such as PHP-FPM use a single process to handle all CGI requests.

nginx/html, and move to that directory by typing `cd /usr/share/nginx/html`. Then, change the ownership attributes of the directory by typing `chown -R nginx owncloud`.

### Create SSL certificates

We're now going to create the SSL certificates that ownCloud needs. Type `cd /etc/nginx/` to move into the configuration directory. Create a directory for the certificates: `mkdir certs`. Now move into that directory: `cd certs`. Type `openssl genrsa -des3 -out server.key 1024` to generate a 1024-bit RSA private key. This command will prompt you for a password. Whatever you choose, please make a note of it. Now type `openssl req -new -key server.key -out server.csr`. This will ask you some identifying questions, but you can hit Return to accept the defaults. Copy the key with `cp server.key server.key.orig`. Convert the key to the format we need by typing `openssl rsa -in server.key.orig -out server.key`. Sign the certificate by typing `openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt`.

### Configure Nginx for ownCloud

We'll fetch an example configuration block from the official documentation. Go to the OwnCloud documentation site ([doc.owncloud.org](http://doc.owncloud.org)). Select Administrators Manual from the side menu. Now enter the term 'nginx' into the search box. One of the search results is the 'Other Web Servers' page and this is what we need. This page contains an Nginx example file. Of the four sections, we just need the SSL section and the WebDAV bits, disregarding the other two. Either cut and paste this into your own Nginx file (or you could create another .conf file to add

```
killermikr@centserver:/home/killermikr
File Edit View Search Terminal Help
Setting up Install Process
Resolving Dependencies
--> Running transaction check
--> Package nginx.i386 0:1.4.1-1.el6.ngx will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package      Arch      Version      Repository      Size
=====
Installing:
nginx        i386      1.4.1-1.el6.ngx    nginx          317 k
Transaction Summary
=====
Install      1 Package(s)

Total download size: 317 k
Installed size: 734 k
Is this ok [y/N]: y
Downloading Packages:
nginx-1.4.1-1.el6.ngx.i386.rpm | 317 kB     00:01
Running rpm_check_debug
Running Transaction Test
```

■ Installing Nginx using YUM

#### TIP LAMP or LEMP?

The most common web hosting setup on Linux is called a LAMP (Linux Apache MySQL PHP), but an Nginx setup is called a LEMP due to the phonetic sound of the name.

to your /etc/nginx/conf.d/ if you prefer). Adapt this to your setup by altering the root string to match our web root (/usr/share/nginx/html). Alter the 'ssl\_certificate' and 'ssl\_certificate\_key' lines so that they match these:

```
ssl_certificate /etc/nginx/certs/
server.crt;
ssl_certificate_key /etc/nginx/
certs/server.key;
```

OwnCloud requires multibyte string support in PHP, so install it by typing `yum install php-mbstring`. Make a final check of the current official documentation on the website to see if there are any further amendments that need

to be made. Restart PHP-FPM with `service php-fpm restart`.

### Create ownCloud database

Create a database and a database user with a password, as with the previous examples.

```
CREATE DATABASE owncloud;
GRANT ALL PRIVILEGES ON owncloud.* TO 'ownuser'@'localhost' IDENTIFIED BY 'your password';
quit;
```

### Complete installation

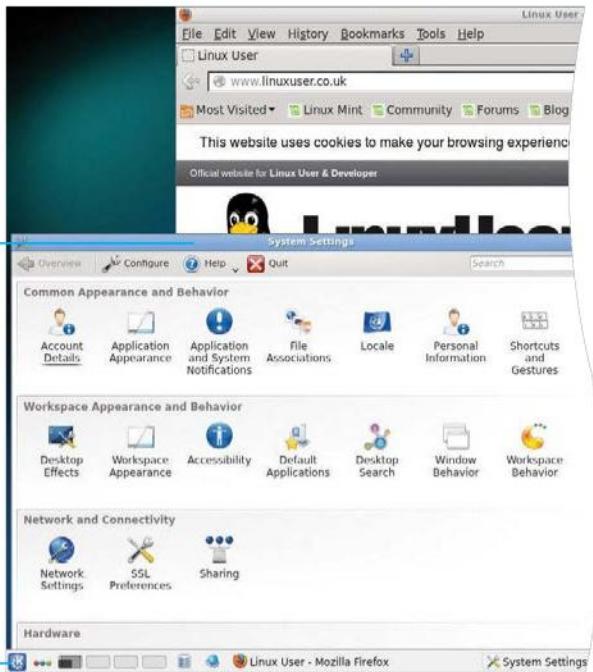
Now browse to `localhost/owncloud` and you should be able to see the ownCloud login page. Create an administrator by specifying a username and password. Click the Advanced tab and specify your database details.

**“Of the four sections, we just need the SSL and the WebDAV bits”**

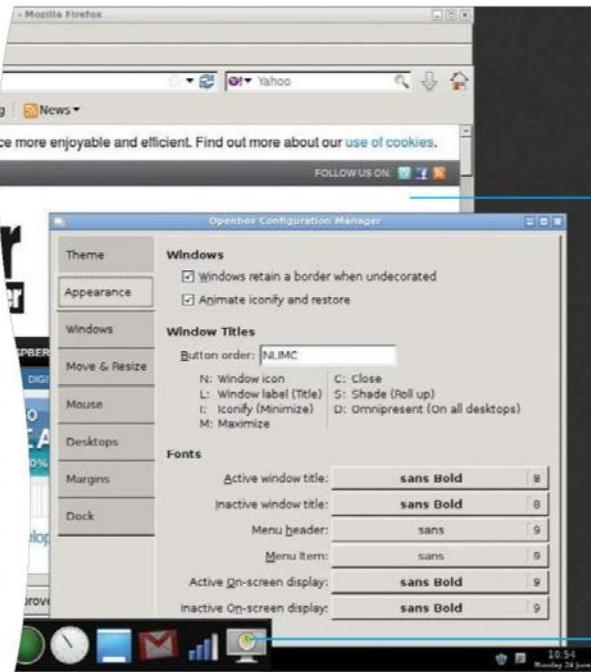


■ Testing that the Nginx installation has worked by pointing the browser at localhost

Speed up a core part of your system by using Openbox instead of the standard window manager



Maintain the other aspects of your favourite desktop environment while using the lighter, faster Openbox



Use Openbox as a desktop environment for super-speedy and customisable workflow

Personalise Openbox to make it more usable in the way you wish with docks, taskbars and extra menus

# Speed up Linux with Openbox

Learn how to install and properly configure the lightweight window manager Openbox and speed up your day-to-day computing without sacrificing usability

**We're always looking for ways to speed up our systems.** Whether we're trying out lighter distros or desktop environments, building from scratch or selecting the perfect array of apps, there are many ways to accomplish this. One of the ways that can sometimes get overlooked, though, is changing your window manager – the set of packages that handles the actual windows of your desktop.

One of the most popular and lighter window managers is Openbox. It's one of the main window managers in LXDE, and readily available to a lot of distros either through their repos or

via the website. It can noticeably speed up your desktop, especially if you're using GNOME, KDE and the like.

Openbox can also be used as your main, supercharged and minimal desktop environment. It uses a much simpler layout than some of the more popular desktop environments; however, it's perfectly usable with a few tweaks and may just greatly increase your workflow.

The best part is, you can go back easily to your old desktop or windows manager whenever, thanks to the way Linux login managers handle desktop sessions.

## Resources

**Openbox:**  
[openbox.org/wiki/Openbox:Download](http://openbox.org/wiki/Openbox:Download)



## 01 Install Openbox

Installing Openbox to your system is very easy. You can find it in your graphical package manager or software centre, or install it with the following for Debian-based systems:

```
$ sudo apt-get install openbox
```

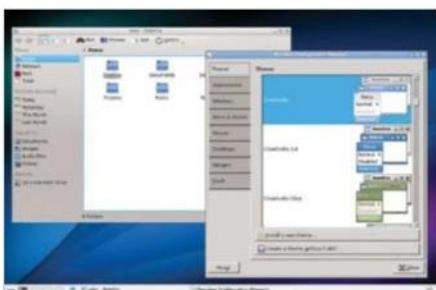
...and for Fedora it's:

```
$ sudo yum install openbox
```



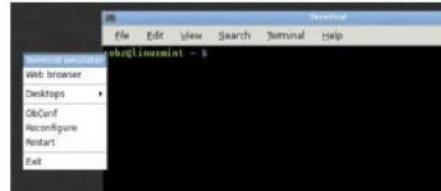
## 02 Use Openbox

Log out of your distro. MDM, GDM, LightDM and KDM will all allow you to select a session at the login screen – open the selection and you'll see that you now have the option to use GNOME/Openbox or KDE/Openbox.



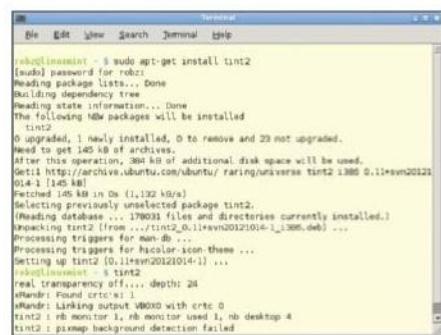
## 03 Basic configure

Openbox is highly configurable, and the most basic configuration can be found in the graphical manager for this. Here you can change the windows theme, the way the windows react during your workflow, and whether or not you want to use a dock.



## 04 Log into Openbox

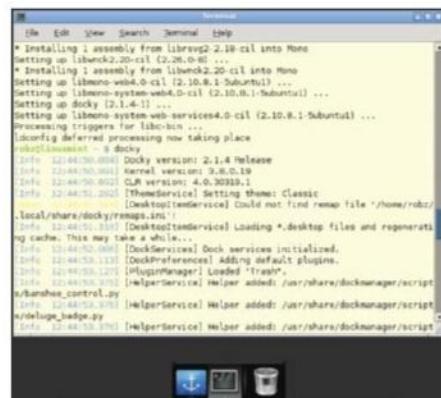
Log back out and select the Openbox desktop from your session manager. After logging back in, you'll be presented with a basic grey desktop and not much more. Right-clicking will open up some options; for now, open the terminal.



## 05 System tray

To get a panel with open windows and system trays, your best bet is to install tint2. To do this, you'll need to simply install it using the terminal we just opened. The package is called tint2, so for Fedora it would be:

```
$ sudo yum install tint2
```



## 06 Docking

You can create a shortcut dock for apps to live on, similar (but better) than what you get in OS X. It was used in Fuduntu, and is nice and lightweight. To install Docky in something like Debian, use:

```
$ sudo apt-get install docky
```

## 07 Desktop compositing

For Docky to work properly, there needs to be some degree of desktop compositing. One of the best ways to do this while still keeping a quick system is to use xcompmgr. Install on Fedora with:

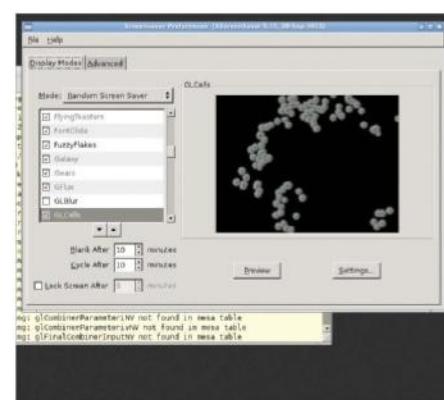
```
$ sudo yum install xcompmgr
```



## 08 Backgrounds

To be able to set a background image and fully customise your Openbox desktop, the best package for the job is Nitrogen. It comes with a graphical interface to choose backgrounds and can be installed with:

```
$ sudo apt-get install nitrogen
```



## 09 Saving screens

You can install a screensaver to Openbox by using the basic xscreensaver. Install it with something like:

```
$ sudo yum install xscreensaver  
xscreensaver-gl
```

To modify it, run `xscreensaver-demo` from the terminal. This also adds power management options.

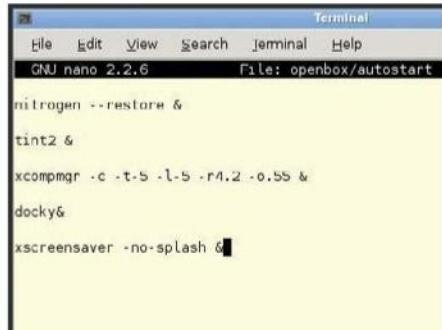
## 10 Autostarting

These will not automatically start when logging into Openbox, so we need to create an autostart script to deal with it. Create a config directory with:

```
$ mkdir ~/.config/openbox
```

...and then open a new autostart file with:

```
$ nano ~/.config/openbox/autostart
```



## 11 Start script

Add the individual elements to the autostart script like so:

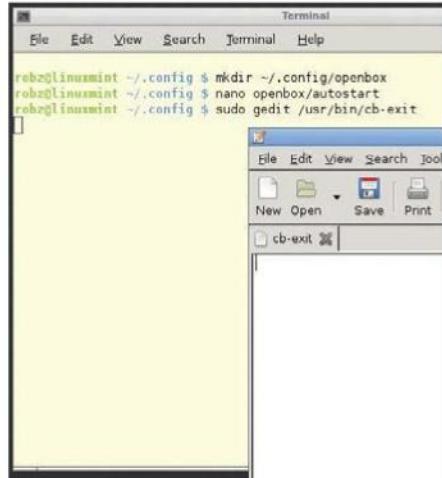
```
nitrogen --restore &
tint2 &
xcompmgr -c -t-5 -l-5 -r4.2 -o.55 &
docky &
xscreensaver -no-splash &
```

Press Ctrl+X and save the script.

## 12 Numlock on

By default, the numlock will not be kept on when logging into Openbox. To get this to happen at startup, install the numlockx package with yum or apt-get, and then add this line to the autostart script:

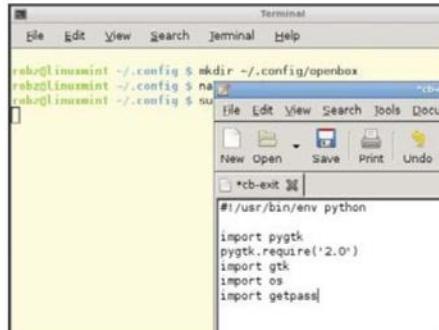
```
numlockx on &
```



## 13 Shut down

Openbox doesn't have a specific menu that lets you shut down graphically. Crunchbang, a Linux distro that uses Openbox, has a great Python script for this that we can borrow from. First of all, create the shutdown menu script with:

```
$ sudo gedit /usr/bin/cb-exi
```



## 14 Python imports

Set up the script so we can use the necessary Python elements with:

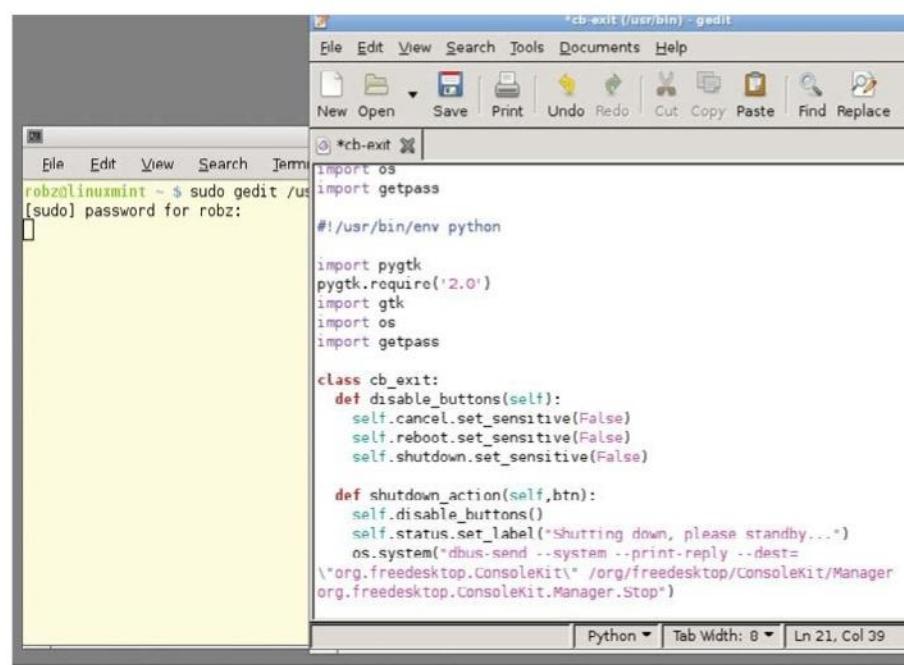
```
#!/usr/bin/env python
```

```
import pygtk
pygtk.require('2.0')
import gtk
import os
import getpass
```

## 15 Shutdown confirmed

For a simple shutdown button, you'll need to do the following in the script:

```
class cb_exit:
    def disable_buttons(self):
        self.shutdown.set_
        sensitive(False)
    def shutdown_action(self,btn):
```



```
self.disable_buttons()
self.status.set_
label("Shutting down, please
standby...")
os.system("dbus-send
--system --print-reply --dest=\"org.
freedesktop.ConsoleKit\" /org/
freedesktop/ConsoleKit/Manager org.
freedesktop.ConsoleKit.Manager.Stop")
def create_window(self):
    self.window = gtk.Window()
    title = "Shut down " +
getpass.getuser() + "? Choose an
option:"
    self.window.set_title(title)
    self.window.set_border_
width(5)
    self.window.set_size_
request(500, 80)
    self.window.set_
resizable(False)
    self.window.set_keep_
above(True)
    self.window.stick
    self.window.set_position(1)
    self.window.connect("delete_
event", gtk.main_quit)
    windowicon = self.window.
render_icon(gtk.STOCK_QUIT, gtk.
ICON_SIZE_MENU)
    self.window.set_
icon(windowicon)
```

```

def create_window(self):
    self.window = gtk.Window()
    title = 'Shut down ' + getpass.getuser() + '? Choose an option:'
    self.window.set_title(title)
    self.window.set_border_width(5)
    self.window.set_size_request(500, 80)
    self.window.set_resizable(False)
    self.window.set_keep_above(True)
    self.window.stick()
    self.window.set_position(1)
    self.window.connect('delete_event', gtk.main_quit)
    windowicon = self.window.render_icon(gtk.STOCK_QUIT, gtk.ICON_SIZE_MENU)
    self.window.set_icon(windowicon)

    #Create HBox for buttons
    self.button_box = gtk.HBox()
    self.button_box.show()

    #Shutdown button
    self.shutdown = gtk.Button(" Shut down")
    self.shutdown.set_border_width(4)
    self.shutdown.connect("clicked", self.shutdown_action)
    self.button_box.pack_start(self.shutdown)
    self.shutdown.show()

    self.vbox = gtk.VBox()
    self.vbox.pack_start(self.button_box)
    self.vbox.pack_start(self.label_box)
    self.vbox.show()
    self.window.add(self.vbox)
    self.window.show()

def __init__(self):
    self.create_window()

def main():
    gtk.main()

if __name__ == "__main__":
    go = cb_exit()
    main()

```

## 16 Shutdown button

That's the function of the button set up; now for the window and button:

```

self.button_box = gtk.HBox()
self.button_box.show()
self.shutdown = gtk.Button(" Shut down")
self.shutdown.set_border_width(4)
self.shutdown.connect("clicked", self.shutdown_action)
self.button_box.pack_start(self.shutdown)
self.shutdown.show()

```

```

#Create HBox for status label
self.vbox = gtk.HBox()
self.vbox.pack_start(self.button_box)
self.status = gtk.Label()
self.status.show()
self.label_box.pack_start(self.status)
self.vbox.show()

self.window.add(self.vbox)
self.window.show()

def __init__(self):
    self.create_window()

```

## 17 Shutdown window

Finally, we finish off the script like so:

```

self.label_box = gtk.HBox()
self.label_box.show()
self.status = gtk.Label()
self.status.show()
self.label_box.pack_start(self.status)

```

## 18 Restart button

To add a reboot button involves almost the same code as the shutdown button. While defining reboot\_action, make it the same as shutdown\_action, but make sure to use the .Restart function from the ConsoleKit. Create the reboot button by simply replacing 'shutdown' with 'reboot' in the same code.



## 19 Cancel button

You can add a cancel button by defining cancel\_action like so:

```

def cancel_action(self,btn):
    self.disable_buttons()
    gtk.main_quit()

```

And then adding the button with:

```

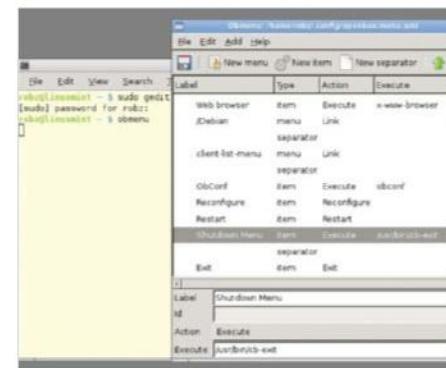
self.cancel = gtk.Button(stock = gtk.STOCK_CANCEL)
self.cancel.set_border_width(4)
self.cancel.connect("clicked", self.cancel_action)
self.button_box.pack_start(self.cancel)
self.cancel.show()

```

## 20 Menu button

To add this shutdown menu to the Openbox menu, you'll need to install obmenu. This graphical tool can help you add apps and scripts to the menu, and is easy to use and very customisable. Install it with something like:

```
$ sudo yum install obmenu
```



## 21 Add button

Open obmenu, and expand the Openbox 3 arrow. Choose a place to add the button and press New Item. Give it any label you wish, such as Shutdown, make sure Action is set to Execute, and set the Execute command to /usr/bin/cb-exit.



## 22 Extra menus

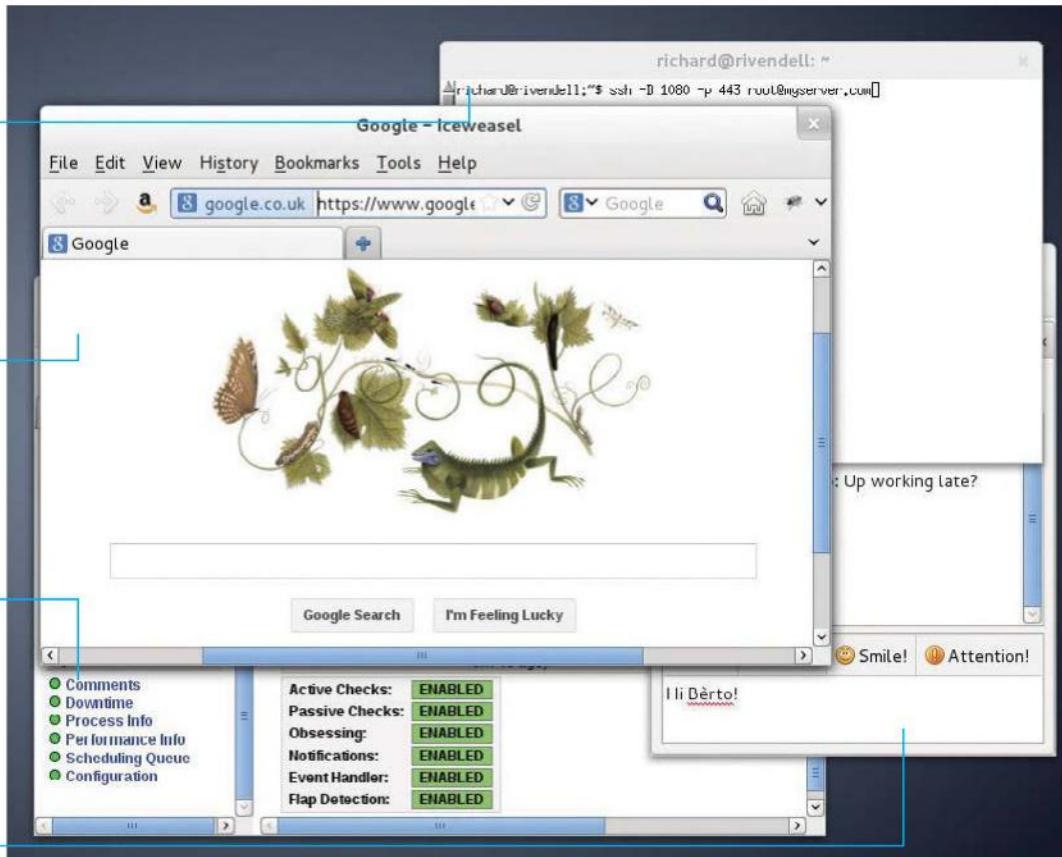
From here you can add extra buttons, apps and functions to the right-click menu and customise your experience. There's a lot of extra customisation you can do with Openbox in general as well, with theming options, behaviour options and much more.

Create a SOCKS proxy server and get your services past firewalls that block the necessary ports

Bypass over-restrictive content filtering by tunnelling your browsing via your server, protecting your unencrypted web traffic from insecure networks, too

Monitor your servers from outside the network without running the corporate VPN client – connect on the fly from any device for out-of-hours monitoring

Need to chat to coders or your office on Jabber when the client site's firewall blocks the XMPP port? Tunnel it over an open port and simply connect your chat client to localhost:8080, for example



# Bypass restrictive firewalls using SSH tunnelling

Create secure network connections on the fly and run safely over insecure networks

If you're still using SSH as just a telnet replacement, you are missing out on borrowing its secure encryption to carry many other network services through insecure Wi-Fi, and overly restrictive firewalls, from wherever you have a laptop or smartphone.

For the bulk of this article, we shall be looking at local port forwarding – the most common and the most useful type – to give secure, VPN-like connections. Why not just use a virtual private network? VPNs aren't always available to you, and some corporate VPNs demand particular client software and configuration, but SSH

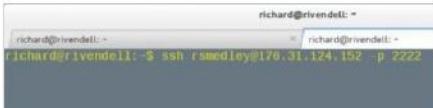
tunnels can always be created on the fly, as and when you need them.

Perhaps you have never read the SSH man page? No? Well, the options you should have been looking at are -L and -R, with a little attention to -N and -f.

Skipping lots of theory, we'll take a practical approach and show you how to use SSH tunnelling in various common scenarios. Read on and find just what these magic switches to the ssh command can do for you, but beware – the power to run rings around firewalls should be used carefully!

## Resources

- SSH client** with SSH daemon on the server
- A server** connected to the internet, preferably with a fixed IP address



## 01 A different port

**1** When you run a normal SSH session, it simply opens an encrypted connection from a spare port on your computer to port 22 on a remote device. For security reasons – many scripts are knocking on port 22 with well-known passwords – you can specify another port.



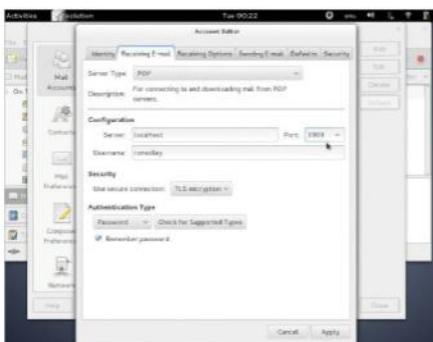
## 02 Insecure access

**Q2** However, inside this encrypted connection you can carry other traffic – hence SSH tunnelling. This means that however insecure your connection (eg cafe Wi-Fi), your traffic is as secure as the level of encryption used by SSH (ie good enough).



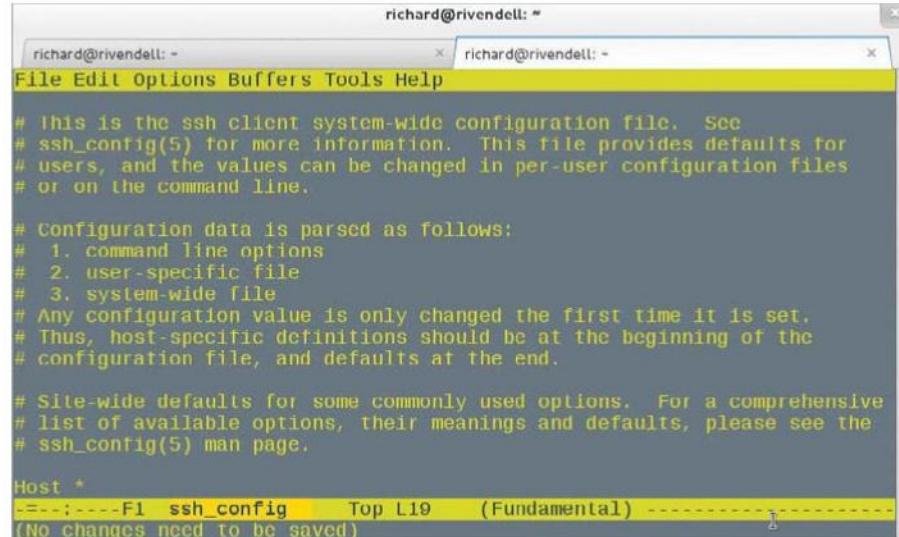
03 Confidential mail

**03** Tunnelling allows you to hide your unencrypted email traffic inside the SSH connection. The -L local-port:host:remote-port creates the tunnel, allowing SMTP (port 25) traffic from the mail-server to appear on (for example) port 3909 locally.



## 04 Local config

**04** Now just configure your email client to connect to port 3909 of the local machine. Localhost and 127.0.0.1 are synonymous, but you



could also use the fully qualified domain name (FQDN) of your local machine. You can do the same for receiving mail via POP.

List of TCP and UDP port numbers - Wikipedia, the free encyclopedia - increased					
Port	Protocol	Description	Source	Official	Notes
5432	TCP	PostgreSQL	MIC CLARION	Official	
5432	UDP	PostgreSQL: A connection-protocol for PostgreSQL	MIC CLARION	Official	
5443	TCP	Sun Grid Engine—Optimizer Service	Sun Grid Engine	Official	
5443	UDP	Sun Grid Engine—Optimizer Service	Sun Grid Engine	Official	
5462	TCP	Yammer Business Solutions - Avant! Remote Control	Yammer Business Solutions	Official	
5462	UDP	Yammer Business Solutions - Avant! Remote Control	Yammer Business Solutions	Official	
5514	TCP	Getting over TLS?	OpenSSL	Official	
5515	TCP	Elipsa RNC Processor [RNC]	Elipsa RNC Processor	Official	
5522	TCP	Secure (anti-virus) based software	Secure (anti-virus) based software	Official	
5523	TCP	Secure (anti-virus) based software	Secure (anti-virus) based software	Official	
5540	UDP	Microsoft Internet Explorer Development & Test Environment	Microsoft Internet Explorer Development & Test Environment	Official	
5540	UDP	default	Microsoft Internet Explorer Development & Test Environment	Official	
5548	TCP	Unlimited Resource Access Netowk (URAN) Network Transfer daemon	URAN	Official	
5549-5552	TCP	spandar-chipperd daemon	spandar-chipperd	Official	
5571	TCP	Windows Live Toolbar client	Microsoft	Official	
5605	TCP	Microsoft Project [MSProject]	Microsoft Project	Official	
5633	TCP	Lotus Notes, Lotus File Transfer Protocol (ESTP)	IBM Lotus	Official	
5646	UDP	Lotus Notes/Lotus File Transfer Protocol (ESTP)	IBM Lotus	Official	
5646-5654	TCP	Microsoft Network Agent [NetAgent]	Microsoft	Official	
5646-5659	TCP	internal Relay Chat [IRC]	internal Relay Chat [IRC]	Official	
5655	TCP	internal Relay Chat [IRC]	internal Relay Chat [IRC]	Official	
5656	TCP	internal Relay Chat [IRC]	internal Relay Chat [IRC]	Official	

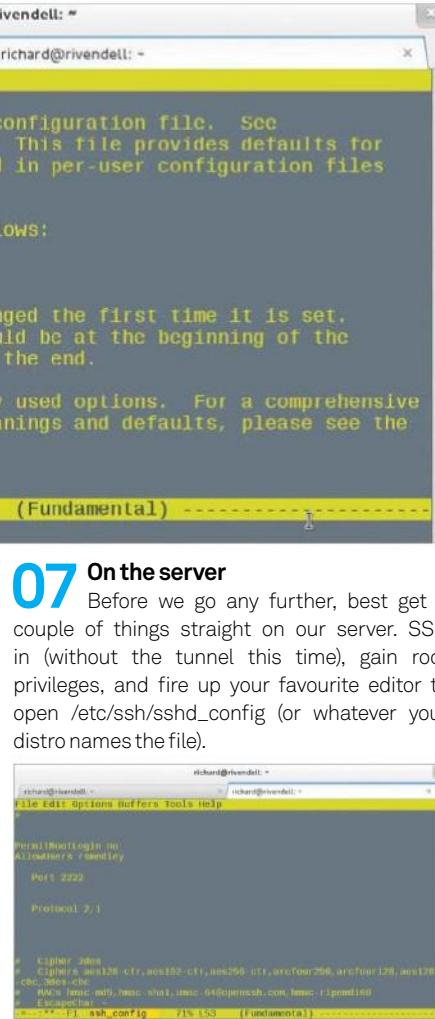
## 05 Pick a number

**Q5** Why port 3909? Port numbers below 1024 are for privileged services. No non-root users should be looking higher than this, but taking a peek at the popular ports in use by other software. Pick a free number such as 6555 or 3989 as your default.



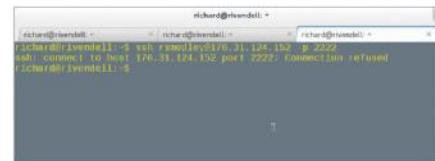
## 06 Two-lane tunnel

While outward-bound SMTP is occasionally blocked, if you're tunnelling for security, best do the incoming POP mail with the same command. As you can see, multiple local tunnels can be expressed in the same ssh command, each with the -L switch.



## 08 Security basics

**88** As well as security settings like a port other than 22, and not allowing root login, here you should uncomment the protocol version 2 setting, so only the more secure protocol version 2 will be used. If both are listed, delete the '1'.



09 Error check

**09** Check you can log in on the new port from another terminal before you close this session! If there is a problem, check that you restarted the SSH server, and typed the correct port and username. If in doubt, return to default port setting.

## 10 Keep yourself alive

While you can add `ServerAliveInterval 60` to your `~/.ssh/config` file, adding `KeepAlive` on the server will work when you connect from other devices or PCs – the `ClientAlive` directives will keep you connected during inactive periods, which is useful for reverse tunnels.

```
richard@rivendell: ~
richard@rivendell: ~
richard@rivendell: ~
richard@rivendell: ~ richard@rivendell: ~ richard@rivendell: ~
File Edit Options Buffers Tools Help
ClientAliveInterval 60
ClientAliveCountMax 99999

RemoteForward 3822 lhost1:22
RemoteForward 3922 lhost2:22
RemoteForward 22880 lhost3:88
RemoteForward 22443 lhost3:443
LocalForward 39801 remotehost:3980
LocalForward 80001 remotehost:8000

ForwardAgent yes
richard@rivendell: ~ richard@rivendell: ~ ssh config 35% L29 (Fund
```

## 11 Keeping track

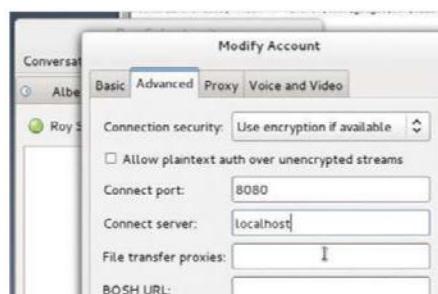
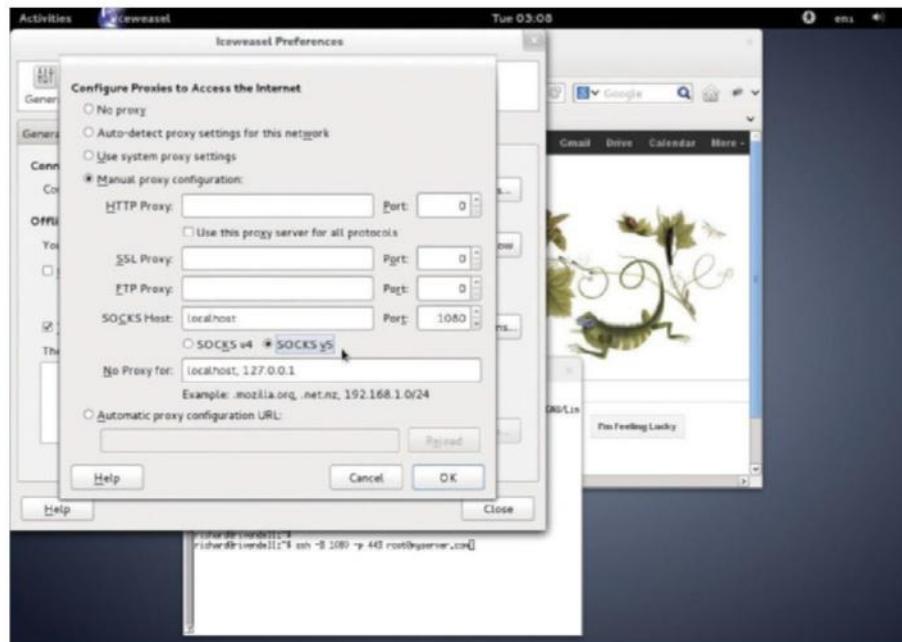
Configs for local and reverse tunnels can also be added at the client end too – handy for keeping track of multiple connections over multiple ports, as well as enabling easier connections from shell scripts. `RemoteForward` = reverse tunnel.

```
richard@x41: ~
richard@x41: ~
richard@x41: ~
richard@x41: ~ ssh -f -i ~/secret_rsa -N rsmed
```

## 12 Switching on

Did you notice those extra switches earlier? `-f` will put SSH in the background before executing a command; `-N` stops the execution of remote commands; `-i` allows you to specify a file for private key, for passwordless connection – other than the standard file locations in `~/.ssh/`

**“Now you can work on remote sites alongside your desktop files. Who needs Dropbox?”**



## 13 Through the firewall

There's much more to SSH tunnelling than keeping your emails from prying eyes. If you're on site and the client company's firewall is blocking ports you need, such as Jabber, set up the tunnel and configure your client to use the appropriate port on localhost.



## 14 Unfiltering content

Similarly, you may find access to a security-related site blocked by overzealous content filtering, and need to tunnel browsing through a machine outside the filter: this time we need to set up a different sort of tunnel, a SOCKS proxy.

## 15 Sock it to me!

`ssh -C -D 1080 -p 443 root@myserver.com`

**-D** is for dynamic port forwarding, creating a SOCKS proxy, over which many services can be carried at once. However, the client applications (such as Firefox), need to be capable of using SOCKS, and need to be configured in the application's preferences.

**1080** is the default port for SOCKS. Others may be tried, but won't work with all software. Get your server to listen on port 443, instead of the non-standard ports we suggested earlier, and you'll find your way unblocked as most firewalls allow 443 for https://.

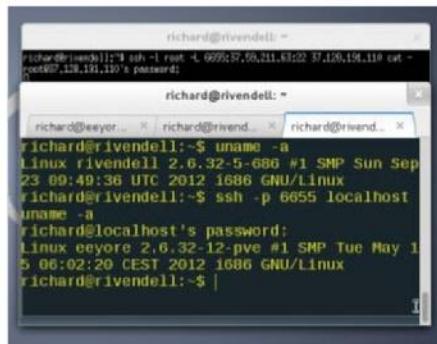
**-C** turns on compression, which speeds up non-binary (ie text) downloading.



## 16 Invisible server

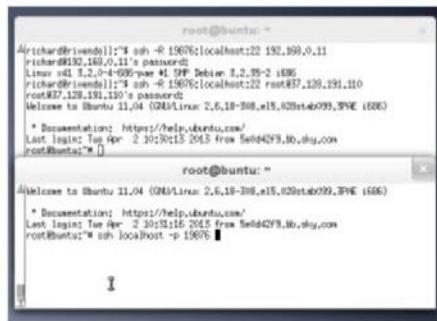
Surprisingly, you may need to tunnel SSH itself through SSH. For example, where the machine we need to reach is invisible to the

```
outside world:
ssh -1 username -L
6655:hiddenmachine:22 gatewayserver
cat -
```



## 17 SSH over SSH

Now we can SSH to the chosen port (6655) on localhost, and we will be executing commands on the hidden server. You can also execute sshin, SCP or SFTP via localhost, port 6655 – tunnelling right through the gateway machine (visible server).



## 18 Power of reverse

A reverse tunnel lets you connect to a NATed machine, without a public IP address. The NATed machine opens a reverse tunnel to a server, and from the server one opens a connection to localhost and the chosen port which connects you back down the tunnel.



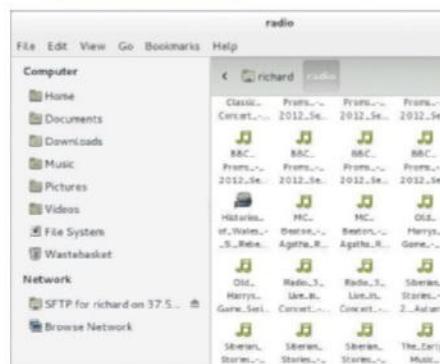
## 19 Third-party access

From a third machine, connect to the server. Then SSH to localhost and you are also connected to the NATed machine. This means from anywhere you can connect to a desktop without an SSH server, if it can run a client.



## 20 GUI help

Some desktop software effectively tunnels through SSH for you, such as your file browser. In Nautilus, go to **File --> Connect to Server** and put in your SSH details. In Konqueror enter **fish://user@server** in the location bar.



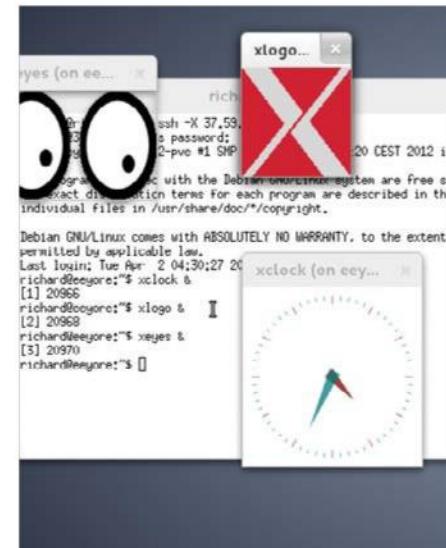
## 21 Drag and drop

Now you can work on remote sites alongside your desktop and locally mounted shares. Who needs Dropbox? Note that as well as SSH, you can do this over FTP or HTTP (WebDAV). GUI-haters can use MC (from the Right menu, select 'shell link'), or mount with SSHFS.



## 22 Remote apps

At its simplest, tunnelling X applications means never having to battle dependencies to install difficult apps on your PC, so long as they're running on a machine to which you have SSH access with X forwarding enabled. In practice, machines on local networks give best (least laggy) results.



## 23 Transcontinental apps

Nevertheless, graphical apps can be run from servers hosted in another country, as long as you are prepared to put up with a little lag in busier apps. You could even browse BBC iPlayer on a UK-hosted box while travelling overseas.



## 24 Remote desktop

Beyond forwarding X apps, we'll have a bit more to say on VNC and remote desktops next month, when we conclude our look at secure remote network apps and look at the more permanent alternative to SSH tunnels – the virtual private network or VPN.

Make a decision from a small selection of ISOs to make the build easier for your environment

**University of Kent UKMIRROR service**

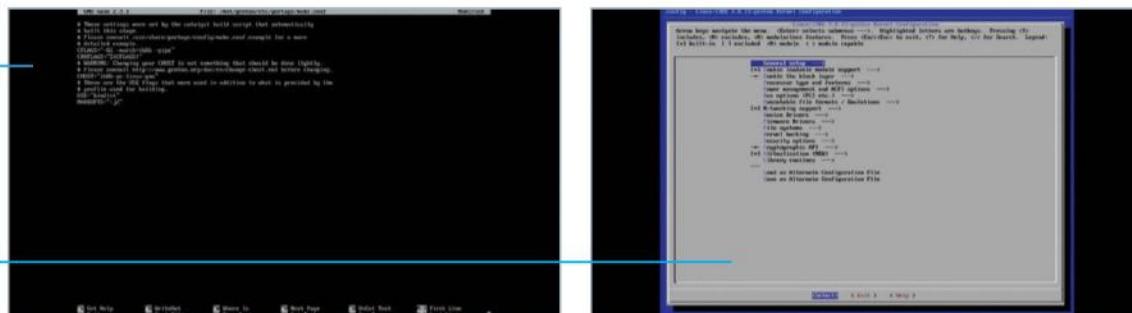
**Index of /sites/www.ibiblio.org/gentoo/releases/x86/current-iso/**

Name	Last modified	Size
<a href="#">Parent Directory</a>	-	-
<a href="#">hardened/</a>	29-May-2013 02:02	-
<a href="#">install-x86-minimal-20130528.iso</a>	28-May-2013 19:44	157M
<a href="#">install-x86-minimal-20130528.iso.CONTENTS</a>	28-May-2013 19:44	3.0K
<a href="#">install-x86-minimal-20130528.iso.DIGESTS</a>	28-May-2013 19:44	732
<a href="#">install-x86-minimal-20130528.iso.DIGESTS.asc</a>	29-May-2013 02:02	1.6K
<a href="#">stage3-i486-28130528.tar.bz2</a>	28-May-2013 19:44	165M
<a href="#">stage3-i486-28130528.tar.bz2.CONTENTS</a>	28-May-2013 19:44	4.0M
<a href="#">stage3-i486-28130528.tar.bz2.DIGESTS</a>	28-May-2013 19:44	716
<a href="#">stage3-i486-28130528.tar.bz2.DIGESTS.asc</a>	29-May-2013 02:02	1.6K
<a href="#">stage3-i686-28130528.tar.bz2</a>	28-May-2013 19:44	165M
<a href="#">stage3-i686-28130528.tar.bz2.CONTENTS</a>	28-May-2013 19:44	4.0M
<a href="#">stage3-i686-28130528.tar.bz2.DIGESTS</a>	28-May-2013 19:44	716
<a href="#">stage3-i686-28130528.tar.bz2.DIGESTS.asc</a>	29-May-2013 02:02	1.6K

Comments or Questions: [help@ukmireservice.org](mailto:help@ukmireservice.org)  
[Follow @UKMireservice](#)  
[Help](#)

Customise your system down to the smallest part to make sure it runs as smooth as possible

Customise the kernel so that there's no bloat in your system



# Create a custom build of Gentoo

Build a custom distro from the ground up to suit your preferences and even speed up your system

Gentoo can be a double-edged sword. It's probably the most customisable Linux distribution available, letting you build it from the ground up to be exactly what you want it to be. However, it's not all that easy to get to grips with and requires some serious skills in Linux to get right.

Once you have, though, Gentoo can be very rewarding. Building packages from source and compiling the kernel yourself feels like you're making the most out of your Linux experience, and can actually help make your system a

lot faster than some of the more popular, preconfigured distros.

In this tutorial, we'll cover taking an image and doing the first-time setup. While we'll be relying on the minimal installation disc and downloading stage tarballs from the internet, there is also a more complete DVD image that you can use to create a more rudimentary setup. A lot of the steps will be the same; however, if you get stuck, the Gentoo website has some great resources and manuals to guide you through any differences.

## Resources

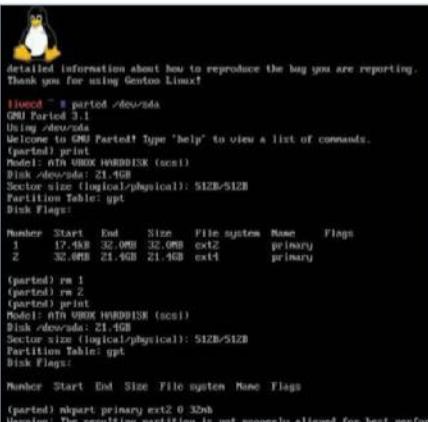
Gentoo live image:  
[www.gentoo.org/main/en/mirrors2.xml](http://www.gentoo.org/main/en/mirrors2.xml)

Name	Last
Parent Directory	
hardened/	29-May-2013 00:00
install-x86_minimal_20130528.iso	28-May-2013 19:44
install-x86-minimal-20130528.iso.CONTENTS	28-May-2013 19:44
install_x86_minimal_20130528.iso.DIGESTS	28-May-2013 19:44
install-x86-minimal-20130528.iso.DIGESTS.asc	29-May-2013 19:44
stage3-i486-20130528.tar.bz2	28-May-2013 19:44
stage3-i486-20130528.tar.bz2.CONTENTS	28-May-2013 19:44
stage3-i486-20130528.tar.bz2.DIGESTS	28-May-2013 19:44
stage3-i486-20130528.tar.bz2.DIGESTS.asc	29-May-2013 19:44
stage3-i686-20130528.tar.bz2	28-May-2013 19:44
stage3-i686-20130528.tar.bz2.CONFLITS	28-May-2013 19:44
stage3-i686-20130528.tar.bz2.DIGESTS	28-May-2013 19:44
stage3-i686-20130528.tar.bz2.DIGESTS.asc	29-May-2013 19:44

**mkpart primary linux-swap 32mb [32 + RAM]mb**

**mkpart primary ext4 [32 + RAM]mb -1s**

Agree to or ignore any prompts. The option at the end of the third command tells Parted to fill up the rest of the disc.



```

Detailed information about how to reproduce the bug you are reporting.

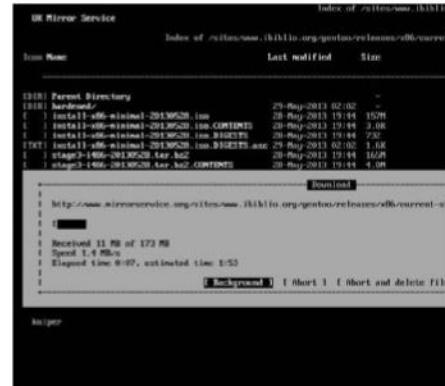
# parted /dev/sda
GNU Parted 3.1
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 21.4GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start End Size File system Name Flags
 1 17.4KB 32.0MB 32.0MB ext2 primary
 2 32.0MB 21.4GB 21.4GB ext4 primary

(parted) mkpart primary ext2 0 32b
Warning: The resulting partition is not properly aligned for best performance.

# swapon /dev/sda2

```



**# date MMDDhhmmYYYY**

...where MM is the month, DD the day/date, etc.

## 07 Take the stage

Move to mount point you just created with `cd`, and then type the following to get a list of mirrors for the stage 3 tarball:

**# links http://www.gentoo.org/main/en/mirrors.xml**

Navigate using the arrow keys to your nearest mirror, go to releases, then your architecture, current stage3, and download a stage3 tarball.

## 08 Extraction

Unpack the tarball you just downloaded with this:

**# tar xvjpf stage3-\*.tar.bz2**

Once it's unpacked, open the configuration file using nano:

**# nano -w /mnt/gentoo/etc/portage/make.conf**

And then we will be able to start some of the initial configuration.

## 09 Make options

The default options already in the configuration file should be good enough for most systems. We can also add an option for how many parallel compilations can occur at once by adding this to the bottom:

**MAKEOPTS="-j[X]"**

...where X is the number of cores you have plus one. Save and exit with Ctrl+X.

## 01 Install and boot

Grab the minimal Gentoo live image and install it to a CD in whatever way you prefer. Restart your system and boot from CD/DVD, and press Enter at the boot screen. You'll then be asked if you want to change your keymap with some option, otherwise it will take you to the live system command line.



## 02 Partitioning

We now need to set up the disc so we can install to it. We'll use Parted. First, check the current layout with:

**# parted /dev/sda**

And then enter `print` in the new Parted command prompt. Use `rm [number]` to remove partitions of that number that you don't need.

## 03 Create partitions

We're going to start from a blank hard drive. We'll need a boot partition, a swap and some space. Use the following three commands:

**mkpart primary ext2 0 32mb**

## 04 Make file systems

Quit out of Parted with `quit`. You'll now need to make the partitions into file systems using the following commands:

**# mkfs.ext2 /dev/sda1**  
**# mkfs.ext4 /dev/sda3**

Create the swap with:

**# mkswap /dev/sda2**

And then swapon with:

**# swapon /dev/sda2**

## 05 Mount partitions

Before we continue, we need to mount the partitions. Do this by first mounting the storage as /mnt/gentoo:

**# mount /dev/sda3 /mnt/gentoo**

Create a boot folder within this:

**# mkdir /mnt/gentoo/boot**

Mount the boot folder:

**# mount /dev/sda1 /mnt/gentoo/boot**

## 06 On time

Make sure the clock is correctly configured to UTC time by simply entering `date`. If it's not, make it UTC time with the following command:

```
# emerge -L /etc/resolv.conf
# These settings were set by the ebuild build script that automatically
# created this file.
# Please consult /usr/share/portage/config/make.conf.example for a more
# detailed example.
#CLASSE="DE_march1606_p1"
#CLASS="DE_march1606_p1"
#WARNING: Changing your CHOST is not something that should be done lightly.
# Please consult http://www.gentoo.org/doc/en/change-chost.xml before changing.
#CHOST="i686-pc-linux-gnu"
# Those are the USE flags that were used in addition to what is provided by the
# profile used for building.
#USE="bindist"
#MODULES="0"
#
```

**10 Prepare build environment**  
We're nearly ready to start building. Save the network/DNS details to the environment with:

```
# cp -L /etc/resolv.conf /mnt/gentoo/etc/
```

Next, mount the /proc file systems, and then bind them to /dev and /sys with:

```
# mount -t proc none /mnt/gentoo/
proc
# mount --rbind /sys /mnt/gentoo/sys
# mount --rbind /dev /mnt/gentoo/dev
```

```
Mount: [SWIPE]
# WARNING: Changing your CHOST is not something that should be done lightly.
# Please consult http://www.gentoo.org/doc/en/change-chost.xml before changing.
# Those are the USE flags that were used in addition to what is provided by the
# profile used for building.
#MODULES="0"
#
```

```
Mount: [SWIPE]
# cp -L /etc/resolv.conf /mnt/gentoo/etc/
# These settings were set by the ebuild build script that automatically
# created this file.
# Please consult /usr/share/portage/config/make.conf.example for a more
# detailed example.
#CLASSE="DE_march1606_p1"
#CLASS="DE_march1606_p1"
#WARNING: Changing your CHOST is not something that should be done lightly.
# Please consult http://www.gentoo.org/doc/en/change-chost.xml before changing.
#CHOST="i686-pc-linux-gnu"
# Those are the USE flags that were used in addition to what is provided by the
# profile used for building.
#USE="bindist"
#MODULES="0"
#
```

**11 Enter build environment**  
The build environment is technically your new Linux environment. We need to make some changes so we can enter it first – basically change the directory we created to root using chroot with:

```
# chroot /mnt/gentoo /bin/bash
# source /etc/profile
# export PS1="(chroot) $PS1"
```

```
# emerge -L /etc/resolv.conf
# These settings were set by the ebuild build script that automatically
# created this file.
# Please consult /usr/share/portage/config/make.conf.example for a more
# detailed example.
#CLASSE="DE_march1606_p1"
#CLASS="DE_march1606_p1"
#WARNING: Changing your CHOST is not something that should be done lightly.
# Please consult http://www.gentoo.org/doc/en/change-chost.xml before changing.
#CHOST="i686-pc-linux-gnu"
# Those are the USE flags that were used in addition to what is provided by the
# profile used for building.
#USE="bindist"
#MODULES="0"
#
```

**12 Portage**  
We need the latest Portage snapshot before we go any further. Emerge it with:

```
# emerge-webrsync
```

This will allow us to install all the packages we need. Update Portage before continuing with:

```
# emerge --sync
```

### 13 Installation profile

You'll now need to choose how to build Gentoo by setting a specific profile. There are three profiles, two of which are of interest to us – Desktop and Server. This will determine the type of packages we use. Set it with:

```
# eselect profile set 2
```

2 is desktop; change it to 3 for a server.

### 14 USEme

The USE variable in make.conf is a powerful tool to configure compiling so it only installs the package support you require. A full list of these flags can be found online or in your system at:

```
# less /usr/portage/profiles/use.desc
```

We'll make our system so it will install files for GNOME and GTK, as well as add ALSA and DVD support. Edit the make file with:

```
# nano -w /etc/portage/make.conf
```

And change USE to:

```
USE="gtk gnome -qt4 -kde dvd alsa"
```

Linux i386 3.8.13-gentoo Kernel Configuration

navigate the menu. <Enter> selects submenus -->. Highlighted <--> excludes. <>> modularizes features. Press <Esc><Esc> to exit. [ ] excluded < > module capable

General setup -->
 [-] Enable loadable module support -->
 [-] Enable the block layer -->
 Processor type and features -->
 Power management and ACPI options -->
 Bus options (PCI etc.) -->
 Executable file formats / Emulations -->
 [\*] Networking support -->
 Device Drivers -->
 Firmware Drivers -->
 File systems -->
 Kernel hacking -->
 Security options -->
 [-] Cryptographic API -->
 [\*] Virtualization (NEW) -->
 Library routines -->

Load an Alternate Configuration File
 Save an Alternate Configuration File

Select < Exit > < Help >

### 15 Kernel time

List the available time zones with:

```
# ls /usr/share/zoneinfo
```

For London, we will do the following:

```
# cp /usr/share/zoneinfo/Europe/London /etc/localtime
```

```
# echo "Europe/London" > /etc/timezone
```

Now it's time to download our kernel. First, get gentoo-sources with:

```
# emerge gentoo-sources
```

Check what kernel version gentoo-sources is pointed at with:

```
# ls -l /usr/src/linux
```

From here, you can start modifying the kernel flags by entering:

```
# cd /usr/src/linux
# make menuconfig
```

Make sure to change only the kernel options you need to. Activate any other required modules. Once done, exit the configuration.

## 16 Compile kernel

The moment of truth – time to compile your kernel. Do this with:

```
# make && make modules_install
```

This will take a while depending on the amount of modules and options you activated. Once it's finished, install the kernel with:

```
# cp arch/x86/boot/bzImage /boot/
kernel-[X]-gentoo
```

...with X being the number we found last step.

## 17 Boot modules

You'll need to set what kernel modules you want to load. To find what modules are available, use:

```
# find /lib/modules/[X]/ -type f
-iname '*.o' -or -iname '*.ko' | less
```

...again where X is the kernel number. You then need to add the modules you want to this file:

```
# nano -w /etc/conf.d/modules
```

## 18 Tabbed file system

We need to set the partitions we created to be mounted properly at boot. Open fstab with:

```
# nano -w /etc/fstab
```

And then set the following options so that the file system we set up works properly:

```
/dev/sda1  /boot  ext2  defaults,noatime 0 2
/dev/sda2  none   swap  sw          0 0
/dev/sda3  /      ext4  noatime   0 1
```

## 19 Networking

You'll need to configure your network for after the reboot. First enter the config file with:

```
# nano -w /etc/conf.d/net
```

And add this like:

```
config_eth0="dhcp"
```

If you're using static IPs, you can add them instead of DHCP. Save, and then make it bootable with:

```
# cd /etc/init.d
# ln -s net.lo net.eth0
# rc-update add net.eth0 default
```

**“Build packages from source and compile the kernel yourself to make the most out of your Linux experience”**

## 20 Root setup

Set the root password with the standard **passwd** command. Now set some basic services by editing:

```
# nano -w /etc/rc.conf
```

Keymaps with:

```
# nano -w /etc/conf.d/keymaps
```

And the hardware clock:

```
# nano -w /etc/conf.d/hwclock
```

If this is not UTC, add **clock="local"** to the file

## 21 Your locale

Specify your locales for the system. A basic setup will need you to edit:

```
# nano -w /etc/locale.gen
```

...and add:

```
en_GB ISO-8859-1
en_GB.UTF-8 UTF-8
```

Save, exit and then type **locale-gen**. Set it as default in **/etc/env.d/02locale** with:

```
LANG="de_DE.UTF-8"
LC_COLLATE="C"
```

And then reload the environment with:

```
# env-update && source /etc/profile
```

## 22 Bootloading

We need to install GRUB so we can boot into Gentoo after a restart. Compile it with:

```
# emerge grub
```

Now create the **grub.conf** file with:

## 23 GRUB code

```
default 0
timeout 15
```

```
title Gentoo Linux
root (hd0,0)
kernel /boot/kernel-3.8.13-gentoo
root=/dev/sda3
```

```
title Gentoo Linux (rescue)
root (hd0,0)
kernel /boot/kernel-3.8.13-gentoo
root=/dev/sda3 init=/bin/bb
```

Make sure to use the correct kernel number. Save and exit.

## 24 GRUB setup and reboot

Create an mtab to install GRUB to with:

```
# grep -v rootfs /proc/mounts > /
etc/mtab
```

And finally, install it with:

```
# grub-install --no-floppy /dev/sda
```

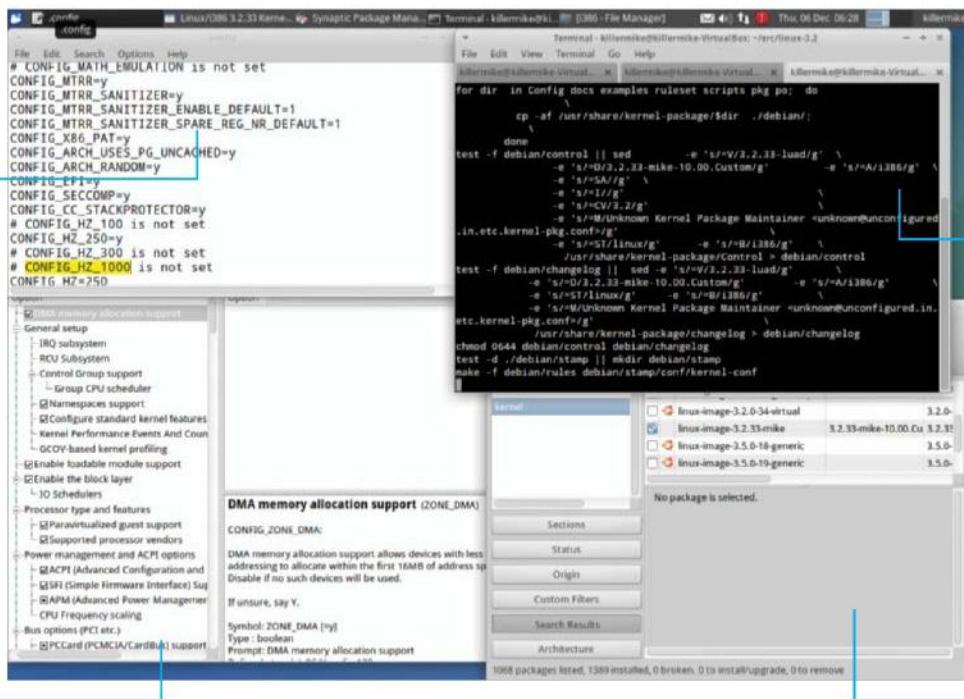
To reboot into your brand new system, exit the chroot and reboot with:

```
# exit
cdimage ~# cd
cdimage ~# umount -l /mnt/gentoo/
dev{/shm,/pts,}
cdimage ~# umount -l /mnt/gentoo{/boot,/proc,}
cdimage ~# reboot
```

# Create a custom Linux kernel to optimise performance

We'll take you through the steps you need to compile your own customised kernel for performance, specialised use and simply to learn how the plumbing works

.config is the file in which you will make most of your changes



Fortunately, there are a few different ways of editing .config, including xconfig, a GUI editor

Compiling the kernel takes a long time, but fortunately, you can leave it running in a terminal window and get on with other work

Once the kernel is complete, you can add it to the system using the standard package tools

**Back in the mid-1990s, recompiling the kernel was something of a necessity, and it was also a good test that a user had mastered the basics of administering Linux.** These days, the stock kernel that comes with most distros has much improved, removing the necessity of kernel recompilation for basic use. However, there are cases where it's well worth becoming familiar with this area of tweaking your system. For one thing, it's a must if you want to access the latest and greatest kernel improvements, hot off the press, so to speak. It's also a good way of understanding how the kernel and other fundamental parts of a Linux system actually

work. It can also be useful when troubleshooting: the newest kernel might bug-fix the problem you're having. On the other hand, an older kernel might be the workaround that you need.

We'll start you off with a simple example that begins with fetching the source archive for the kernel that you are currently running, proceeding through to configuration, compilation and installation. Following this, we'll go through some examples that are a bit more specialist. Most of the examples are for Debian-derived distros, but we've deliberately kept things as neutral as possible and added some notes for how to handle things on Red Hat-based distros such as Fedora.

```
Reading state information... Done
Building dependency tree
The following extra packages will be installed:
dh-aparmor dh-aparmor-dbg git git-man libhtml-test
intool-debian libalgroithm-diff-perl libalgroithm-difftest-perl
libalgroithm-merge-perl libbdg-perl libdw liberror-perl libgettextpo
liblzo2-dev liblzma5-dev liblzo2-2.0.4-6-dev libsys-hostname-long-perl
libtinfo-dev libxattr1-dev po-debconf
Selected packages:
kernel-patch-lkd dmutils dh-nuke debian-keyring g++-multilib
git g++-multilib gcc-4.4-doc liblzo2-2.0.4-6-dev libtinfo-dev git-damn-run
git-damn-run liblzo2-2.0.4-6-dev git liblzo2-2.0.4-6-dev libtinfo-dev
git-gui gitk gitweb ncurses-term liblzo2-2.0.4-6-dev liblzo2-2.0.4-6-dev
The following NEW packages will be installed:
dmutils-dev libcrash-essential crash dh-aparmor dh-aparmor-dbg fakeroot
git g++-4.6 git-core git-damn-run liblzo2-2.0.4-6-dev libtinfo-dev
git-damn-run liblzo2-2.0.4-6-dev liblzo2-2.0.4-6-dev liblzo2-2.0.4-6-dev
liblzo2-2.0.4-6-dev liblzo2-2.0.4-6-dev liblzo2-2.0.4-6-dev liblzo2-2.0.4-6-dev
After this operation, 88.4 MB of additional disk space will be used.
Do you want to continue [Y/n]? 
```

## 01 Install prerequisites

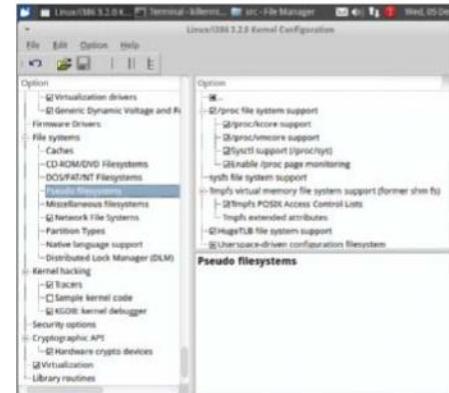
Begin by fetching the tools needed to create a suitable build environment. Enter 'sudo apt-get install fakeroot crash kexec-tools makedumpfile kernel-wedge git-core libncurses5 python libncurses5-dev kernel-package libelf-dev binutils-dev' followed by 'sudo apt-get build-dep linux-image'.

```
Terminal - killermike@killermike-VirtualBox: ~/linux-3.2.0
File Edit View Terminal Go Help
Reading state information... Done
Building dependency tree
The following extra packages will be installed:
dh-helper dh-aparmor dh-aparmor-dbg git git-man libhtml-test
intool-debian libalgroithm-diff-perl libalgroithm-difftest-perl
libalgroithm-merge-perl libbdg-perl libdw liberror-perl libgettextpo
liblzo2-dev liblzma5-dev liblzo2-2.0.4-6-dev libsys-hostname-long-perl
libtinfo-dev libxattr1-dev po-debconf
Selected packages:
kernel-patch-lkd dmutils dh-nuke debian-keyring g++-multilib
git g++-multilib gcc-4.4-doc liblzo2-2.0.4-6-dev libtinfo-dev git-damn-run
git-damn-run liblzo2-2.0.4-6-dev git liblzo2-2.0.4-6-dev libtinfo-dev
git-gui gitk gitweb ncurses-term liblzo2-2.0.4-6-dev liblzo2-2.0.4-6-dev
The following NEW packages will be installed:
dmutils-dev libcrash-essential crash dh-aparmor dh-aparmor-dbg fakeroot
git g++-4.6 git-core git-damn-run liblzo2-2.0.4-6-dev libtinfo-dev
git-damn-run liblzo2-2.0.4-6-dev liblzo2-2.0.4-6-dev liblzo2-2.0.4-6-dev
liblzo2-2.0.4-6-dev liblzo2-2.0.4-6-dev liblzo2-2.0.4-6-dev liblzo2-2.0.4-6-dev
Using defaults found in /boot/config-3.2.0-34-generic
Configuration written to .config
killermike@killermike-VirtualBox: ~/linux-3.2.0
```

## 04 Generate a .config file

The (hidden) file '.config', located in the source code directory, tells the compiler what to build. The configuration file for each installed kernel is stored in the /boot directory, but you can capture the configuration of the current kernel (a good starting point) by typing 'make oldconfig'.

It's a specialist feature and mainly used by developers. You can use xconfig for this. Set 'CONFIG\_DEBUG\_INFO' to 'n'.



## 07 Invoke xconfig

Type 'make xconfig' to launch the GUI config file editor. It's a good way to gain an overview, and it offers information for most of the options. Run 'sudo apt-get install libqt4-core libqt4-dev libqt4-gui' if it complains about not being able to find Qt.

## 08 Prepare Debian scripts

Some required scripts lose their execution privileges due to how apt-get works. Rectify this by typing 'chmod -R u+x debian/scripts/\*' and then 'chmod u+x debian/rules'.

```
Terminal - killermike@killermike-VirtualBox: ~/src
File Edit View Terminal Go Help
patching file ubuntu/dm-raid4-5/dm-memcache.h
patching file ubuntu/dm-raid4-5/dm-raid4-5.h
patching file ubuntu/dm-raid4-5/dm-raid45.h
patching file ubuntu/dm-raid4-5/Kconfig
patching file ubuntu/dm-raid4-5/dm-region-hash.h
patching file ubuntu/dm-raid4-5/ROM
patching file ubuntu/dm-raid4-5/dm-raid4-5.c
patching file ubuntu/dm-raid4-5/dm-message.c
patching file ubuntu/dm-raid4-5/Makefile
patching file ubuntu/dm-raid4-5/dm-message.h
patching file ubuntu/dm-raid4-5/dm-memcache.c
patching file ubuntu/dm-raid4-5/dm-region-hash.c
patching file security/Kconfig
patching file security/inode.c
patching file security/capability.c
patching file security/Makefile
patching file security/device_cgroup.c
patching file security/commoncap.c
patching file security/security.c
patching file security/yama/Kconfig
patching file security/yama/Makefile
patching file security/yama/lsm.c
patching file security/integrity/ima/ima_api.c
patching file security/integrity/ima/Kconfig
```

## 09 Recreating the Source Tree

If you mess things up and want to start from scratch, delete the source directory. Backup your .config file first, if needed. Then run "tar xz!" on the source archive to unpack it. Move into the directory and type "zcat ../[name of diff archive] | patch -p1" to add the Ubuntu patches into the source tree.

## 02 Fetch the kernel (source archive)

If you need the latest kernel, use Git to fetch it (see later step), but we are going to use the standard package tools in the first example. Use 'apt-get source linux-image-\$(uname -r)' to install the source for the currently running kernel.



## 03 Examine the source directory

You should now have a source directory in the current directory. Move into it using the cd command. Note that there is an archived (tar.gz) copy as well. In addition, there is a diff file that contains the Ubuntu-specific additions to the standard kernel source tree.

## 05 Edit .config

Open up .config in a text editor. Note that there are thousands of options, and this approach is best if you know exactly what settings you would like to edit. It's a good idea to search for 'CONFIG\_LOCALVERSION' to add a small identifying string for your custom kernel.

The screenshot shows the 'xconfig' window with the 'Options' tab selected. It lists various kernel configuration options under categories like 'Capacitors to digital converter', 'Digital to analog converter', 'Direct Digital Synthesis', 'Digital gyroscopes sensors', 'Hardware Analyzer', 'Imperial', 'Infrared remote control', 'Light sensors', 'Magnetometer sensors', 'Active energy metering IC', 'Resistor to digital converter', 'Speaker tonality speech', 'Speaker volume speech', 'Software-defined Radio', 'System Infrastructure Remote IO', 'System Platform Specific Device ID', 'Hardware Splice drivers', 'GPMC/I2C Hardware Support', 'Virtualization Drivers', 'File Systems', and 'Documentation'. A specific option, 'CONFIG\_DEBUG\_INFO=y', is highlighted with a blue selection bar.

## 06 Turn off debugging

One way to speed things up and produce smaller files is to turn off debugging.

## 10 Compile the kernel

Tell the build environment how many cores you want to use with ‘export CONCURRENCY\_LEVEL=[number of cores]’. Begin the build process with ‘fakeroot make-kpkg --initrd --append-to-version=--lud kernel-image kernel-headers’, which takes about an hour on a Core Duo 2.7GHz system.

## 11 Install kernel

Warning: This is stage where you actually make some changes to your machine. Using our method means that we can install our custom kernel just like any other package. Type ‘sudo dpkg -i linux-image[version].deb’ and then, ‘sudo dpkg -i linux-headers-[version].deb’.

## 12 Edit GRUB 2

Open up /etc/default/grub in a text editor. Edit the line ‘GRUB\_HIDDEN\_TIMEOUT=0’ to remove the zero. Now run ‘update-grub’. The Grub Menu will be displayed at startup. The old, stock

kernel can be found under the ‘Previous Linux Versions’ entry. On reboot, type ‘uname -a’ to check the kernel version.

## 13 Fetch the kernel using Git (Ubuntu)

This gives you the latest version of the Ubuntu kernel source. Install Git: ‘sudo apt-get install git-core’. Fetch the source with ‘git clone git://kernel.ubuntu.com/ubuntu/ubuntu-oneiric.git’. Substitute ‘oneiric’ with the official distribution name that you are using.

## 14 Fetch kernel manually

Now that you’re familiar with compiling the kernel, consider grabbing a newer kernel from [kernel.org](#), although these kernels don’t have the extra additions of Ubuntu. Unpack the kernel with ‘tar xjfv linux.<version>.tar.bz2’ and repeat the earlier steps.

## 15 Building the kernel on Red Hat

Many of the techniques already applied can be used with Red Hat-derived distributions such as Fedora. Begin by installing the prerequisites: ‘su -c ‘yum install rpmdevtools yum-utils’’. Then type, ‘rpmdev-setuptree’, and then ‘yumdownloader --source kernel’ to fetch the kernel. ‘su -c ‘yum-builddep kernel-[use tab completion].src.rpm’’ and then ‘rpm -Uvh kernel-[use tab completion].src.rpm’. Move into the SPECS directory with ‘cd ~/rpmbuild/SPECS’ and type ‘rpmbuild -bp --target=\$(uname -m) kernel.spec’. Now cd into the source directory (something like, rpmbuild/BUILD/kernel-3.6.fc17/linux-3.6.8-2.fc17.i686/)

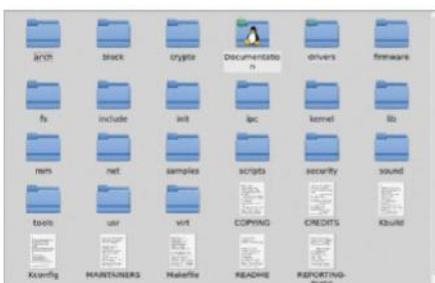
## 16 Begin the build (Red Hat)

Generate and edit the .config file as before. Open ~/rpmbuild/SPECS/kernel.spec and change the line '#% define buildid' from '.local' to something unique to identify your kernel. From within, type ‘~/rpmbuild/SPECS’ then ‘rpmbuild -bb --without debug debuginfo --target=‘uname -m’ kernel.spec’ to build.

## 17 Install ccache

ccache is a compiler cache that is particularly useful when compiling the kernel, although you only start to see the full benefit on repeat compilations. Enter ‘sudo apt-get install ccache’. Then type ‘export CC=“ccache gcc”’ and then ‘export CXX=“ccache g++”’.

**18 Cross-compilation (Raspberry Pi)** You can use a powerful machine to ‘cross-compile’ a kernel for a smaller one. We’ll use the Raspberry Pi as an example. You will need a way of accessing files on the Pi storage device. Type ‘sudo apt-get install gcc-arm-linux-gnueabi make git-core ncurses-dev’ on the PC.



## 19 Fetch the source

On the PC, create a directory and then move into it. Then type ‘git clone https://github.com/raspberrypi/firmware’ followed by ‘git clone https://github.com/raspberrypi/linux.git’. Now move into the ‘linux’ directory.

## 20 Configuration

You can fetch the config from a running Raspberry Pi by typing ‘sudo zcat /proc/config.gz > .config’ into a shared directory, and then copying it to the ‘linux’ directory on the PC. Type ‘make ARCH=arm CROSS\_COMPILE=/usr/bin/arm-linux-gnueabi- oldconfig’.

```
mousedev: PS/2 mouse device common for all mice
sd 0:0:0:0: [sda] Attached SCSI disk
TCP cubic registered
NET: Registered protocol family 17
UFP support v0.3: implementor 41 architecture 1 part 20 variant b rev 4
input: AT Raw Set 2 keyboard as /devices/fpga:06/serio0/input/input0
input: ImExPS/2 Generic Explorer Mouse as /devices/fpga:07/serio1/input/input1
EXT3-fs (sda2): error: couldn't mount because of unsupported optional features (240)
EXT2-fs (sda2): error: couldn't mount because of unsupported optional features (240)
EXT4-fs (sda2): mounted filesystem with ordered data mode. Opts: (null)
VFS: Mounted root (ext4 filesystem) readonly on device 8:2.
devtmpfs: mounted
Freeing init memory: 132K
INIT: version 2.88 booting
[info] Using makefile-style concurrent boot in runlevel S.
[....] Starting the hotplug events dispatcher: udevdudev[134]: starting version
175
[ok]
[ok] Synthesizing the initial hotplug events...done.
[ok] Waiting for /dev to be fully populated...done.
Starting fake huclock: loading system time.
Sun Oct 28 22:59:21 UTC 2012
[....] Setting preliminary keymap...
```

# “The old, stock kernel can be found under the ‘Previous Linux Versions’ entry”



## 22 Tweak for MIDI performance 1

Here’s an example to tweak the kernel for improved MIDI latency for an AV workstation. First, fetch a patch archive at an appropriate level for your kernel from [www.kernel.org/pub/linux/kernel/projects/rt/](http://www.kernel.org/pub/linux/kernel/projects/rt/). Repeat the procedure for recreating the source tree, but apply the RT patch before the Ubuntu one.

```
File Edit View Terminal Go Help
killermike@killermike-desktop: ~/pi
CC init/do_mounts_rtd.o
CC init/do_mounts_initrd.o
CC init/do_mounts_md.o
LD init/mounts.o
CC init/intrtrans.o
CC init/calibrate.o
LD init/built-in.o
HOSTCC usr/gen_init_cpio
GEN usr/initramfs_data.cpio
S usr/initramfs_data.o
LD usr/built-in.o
CC arch/arm/kernel/elf.o
AS arch/arm/kernel/entry-armv.o
AS arch/arm/kernel/entry-common.o
CC arch/arm/kernel/irq.o
CC arch/arm/kernel/process.o
CC arch/arm/kernel/prtrace.o
CC arch/arm/kernel/signal.o
CC arch/arm/kernel/setup.o
CC arch/arm/kernel/signal.o
arch/arm/kernel/sys_arm.o
arch/arm/kernel/stacktrace.o
CC
```

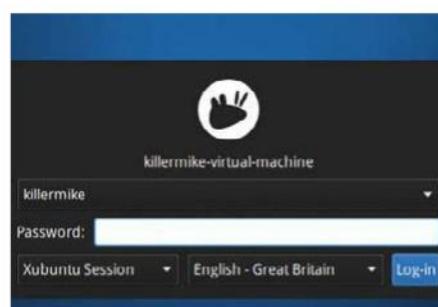
```
.config
# CONFIG_MATH_EMULATION is not set
CONFIG_MTRR=y
CONFIG_MTRR_SANITIZER=y
CONFIG_MTRR_SANITIZER_ENABLE_DEFAULT=1
CONFIG_MTRR_SANITIZER_SPARE_REG_NR_DEFAULT=1
CONFIG_X86_PAT=y
CONFIG_ARCH_USES_PG_UNCACHED=y
CONFIG_ARCH_RANDOM=y
CONFIG_EFI=y
CONFIG_SECCOMP=y
CONFIG_CC_STACKPROTECTOR=y
# CONFIG_HZ_100 is not set
CONFIG_HZ_250=y
# CONFIG_HZ_300 is not set
# CONFIG_HZ_1000 is not set
CONFIG_HZ=250
CONFIG_SCHED_HRTICK=y
CONFIG_KEXEC=y
CONFIG_CRASH_DUMP=y
CONFIG_KEXEC_JUMP=y
CONFIG_PHYSICAL_START=0x1000000
```

## 21 Cross-compile kernel

You can then edit the config with ‘make ARCH=arm CROSS\_COMPILE=/usr/bin/arm-linux-gnueabi- xconfig’. Now run ‘make ARCH=arm CROSS\_COMPILE=/usr/bin/arm-linux-gnueabi- -k’ to compile. Place the new kernel (linux/arch/arm/boot/Image /boot/kernel.img) in /boot/kernel.img.

## 23 Tweak for MIDI performance 2

Follow the steps as for a normal installation, but first carry out some tweaks in the .config file. Ensure that ‘CONFIG\_HZ\_1000’ is set to ‘y’ and that ‘CONFIG\_APM’ is set to ‘n’.



## 24 Work inside a virtualiser

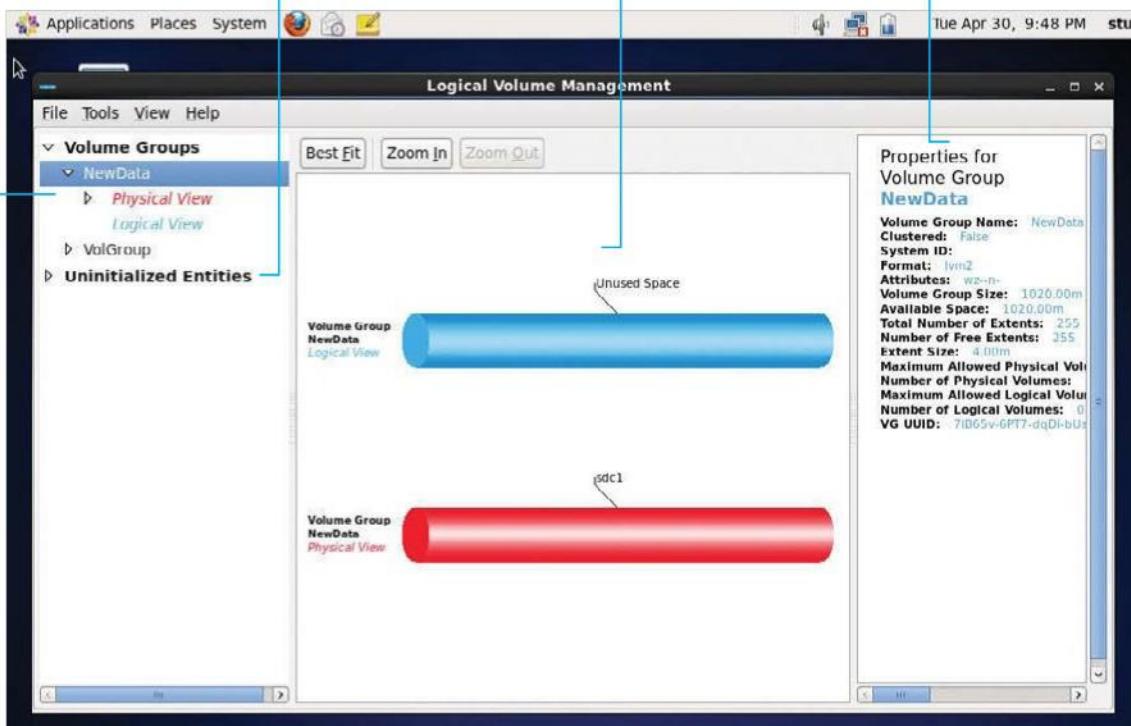
If this is the first time you’ve built a kernel, you might like to start by working inside a virtualiser (eg QEMU). This gives numerous advantages such as being able to pause the build process. Try to devote as much memory and as many cores as you possibly can.

This gives the make-up of the volume groups and the breakdown of them. It's much more visually appealing than the command-line version

Here you will find all unused members (disks) that can be added into the volume groups and expand the disk space

If you select a volume group, this central visual gives you the make-up of the VG with logical and physical disks

This column gives you advanced information on the make-up of the volume group. It gives some useful information such as size and system attributes



# Resize your disks on the fly with LVM

Grow your disks like a true master and never reformat and restore your drive again

**Often we see or hear of people running out of disk space on their Linux systems and resorting to resizing tools or worse.** Linux has the ability to extend (or shrink) logical partitions across spare space or even across additional disks.

This guide walks you through the why and the how of resizing your disks.

We will be creating an additional new LVM-based partition on our virtual CentOS server, as well as the underlying structures that are called physical volumes (PVs) and volume groups (VGs) that make this awesome technology possible.

These three different parts together make up LVM (Logical Volume Manager). The best way to think about LVM setups is that they are like a layered sponge cake. On the bottom layer we have the physical volumes: the hard disks. Then we have the volume groups, the cake's 'cream' that can in effect cement the disks together and provide a smooth contiguous surface to carve out our partitions. This means that a partition can be bigger than a single disk. Also, when you want to expand a disk, as long as you have space in your volume group, you can.

## Resources

VirtualBox: [www.virtualbox.org](http://www.virtualbox.org)

CentOS 6.4 ISO: [wiki.centos.org](http://wiki.centos.org)

```
root@localhost:~ [stu@localhost Desktop]$ su -
Password:
[root@localhost ~]# lvs
  LV   VG     Attr   LSize Pool Origin Data%  Move Log Cpy%Sync Convert
  lv_root VolGroup -wi-ao--- 13.54g
  lv_swap VolGroup -wi-ao--- 1.97g
[root@localhost ~]#
```

## 01 Get all your disks in a row

Essentially LVM separates the logical volumes from the physical disks with a glue in between called Volume Groups (or VGs for short).

Firstly, you may actually be using LVM without realising it! Most newer distros use LVM when they do their install. To see if you are, just open a root console and type `lvs`.

If you see something like that shown in the picture above, then you are already using LVM.

In our example we will be creating a new logical volume from scratch, but the same principles around resizing and reducing still apply.

## 02 Building our new LVM setup

Use the command `fdisk -l` and find the disk we're going to use as the first disk in our LVM setup. In our example, sdb is our new disk.

To tell Linux that the disk is going to be an LVM disk, we need to set the disk type (Label) to LVM. Use the command: `fdisk /dev/sdb`

Press 'n' to add a new partition. The type of disk we are adding is a primary, so press 'p' and as we are only putting a single partition on the disk, we can select partition number 1. Accept the geometry the machine suggests by pressing

Enter on the questions. Don't quit out of fdisk yet as we have more to do!

```
root@localhost:~ [stu@localhost Desktop]$ su -
Password:
[root@localhost ~]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xdd31f2d9.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help):
```

## 03 Preparing the bottom layer of our LVM cake: physical volumes

Once we have set up the partition, we need to identify it as an LVM device. To do this while still in fdisk, press 't' to change the disk label, and enter the label ID, which in our case is 8e. Follow

**“Think of LVM setups being like a layered sponge cake”**

this by 'w' (to write the changes to disk). At this point we have created a partition and identified it as an LVM disk type. So now we need to 'bless' or prepare the physical disk upon which LVM will be based. To do this we use a command called `pvcreate`. This effectively prepares and marks the disk as an LVM disk. To do this, use the command: `pvcreate /dev/sdb1`

```
root@localhost:~ [stu@localhost Desktop]$ su -
Password:
[root@localhost ~]# pvcreate /dev/sdb1
  Physical volume "/dev/sdb1" successfully created
[root@localhost ~]#
```

## 04 Followed by the LVM jam: volume groups

The bottom 'cake layer' is now created and we can make the jam in the middle: the volume groups. Each of these is like a chunk of storage that you can slice into one or several logical volumes. The difference is that you can expand and reduce disks within the volume group. The `vg_data` bit is the name for the volume group: `vgcreate vg_data /dev/sdb1`

If you want to span multiple disks, just add in the extra devices after `/dev/sdb1`. Remember to mark them as LVM disks using the `pvcreate` command and then marking them as LVM disks in fdisk!

```
root@localhost:~ [stu@localhost Desktop]$ su -
Password:
[root@localhost ~]# vgcreate vg_data /dev/sdb1
  Volume group "vg_data" successfully created
[root@localhost ~]#
```

## 05 Adding the top of the cake: logical volumes

Now that we have created the second layer, we can add the logical volumes:

`lvcreate -L 1G -n lv_data vg_data`

This command will create a 1GB logical volume

```
root@localhost:~ [stu@localhost Desktop]$ su -
Password:
[root@localhost ~]# lvcreate -L 1G -n lv_data vg_data
  Logical volume "lv_data" created
[root@localhost ~]#
```

## “Remember, a VG is just a chunk of space”

from the volume group we created. To clarify, lv\_data is our new logical volume, while vg\_data is the volume group to take it from. You can make the volume as small or large as you like, given a big enough volume group. We are using the entire volume group, rather than just a small portion of it. All the volume groups can be found under the /dev/mapper directory.

Now we can create a logical disk on top of it.

```
root@localhost:~#
File Edit View Search Terminal Help
[root@localhost ~]# mkfs.ext4 /dev/mapper/vg_data-lv_Data
mke2fs 1.41.12 (17-May-2010)
Filesystem label:
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=8 blocks, Stripe width=8 blocks
65536 inodes, 262144 blocks
13107 blocks (5.00% reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98384, 163840, 229376

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information:
This filesystem will be automatically checked every 28 mounts
180 days, whichever comes first. Use tune2fs -c or -i to
[root@localhost ~]#
```

## 06 Making our volumes

Now we can format the disk using the following command:

`mkfs.ext4 /dev/mapper/vg_data-lvdata`  
To test the partition, try mounting it.

Now this is where things get interesting. This could be a 60GB or 600GB disk. We just chose the 1GB as an example. Now, say we have filled this disk with all our ISOs and such, how do we fix it and add that shiny new disk into the logical disk? This is where we see the joy of LVM. We can expand our volume over the new disk and make our volume bigger!

## 07 Expanding our volumes

To do this, it is a very similar method to creating an LVM; we need to in effect prepare the new disk as an LVM member. To do so, repeat the process we performed earlier to prepare the disks (with fdisk and pvcreate).

```
File Edit View Search Terminal Help
[root@localhost ~]# resize2fs /dev/mapper/vg_data-lv_Data
resize2fs 1.41.12 (17-May-2010)
Resizing the filesystem on /dev/mapper/vg_data-lv_Data to 524288 (4k) blocks.
The filesystem on /dev/mapper/vg_data-lv_Data is now 524288 blocks long.
[root@localhost ~]#
```

```
File Edit View Search Terminal Help
[root@localhost ~]# lvextend -L +1G /dev/mapper/vg_data-lv_Data
Extending logical volume lv_Data to 2.00 GiB
Logical volume lv_Data successfully resized
[root@localhost ~]#
```

Now that the partition is prepared, we can add it into the volume group we just created. To do this we use the vgextend command, as shown below. The syntax is quite straightforward with the first argument being the volume group you want to add the disk to and secondly, the disk you wish to add.

`vgextend vg_data /dev/sdc1`

Now that the VG is extended, we need to perform a similar process to increase the logical disk to use the underlying storage we expanded before. Below is a simple example to just expand our VG with a 1GB partition.

`lvextend -L +1G /dev/mapper/vg_data /dev/sdc1`

## 08 Generation game

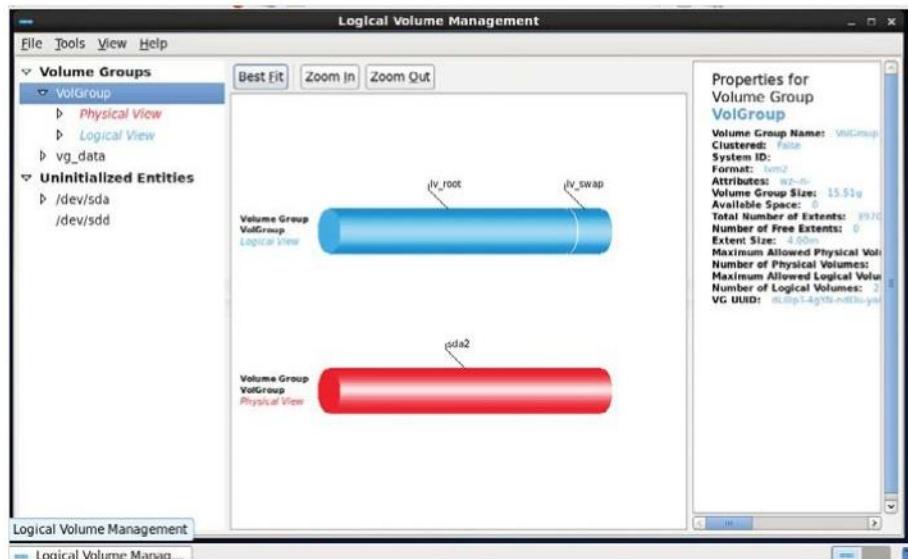
And lastly we will need to extend the actual file system. There is a command called `resizesfs` that we can use. As the name implies, it is used to resize the filesystem.

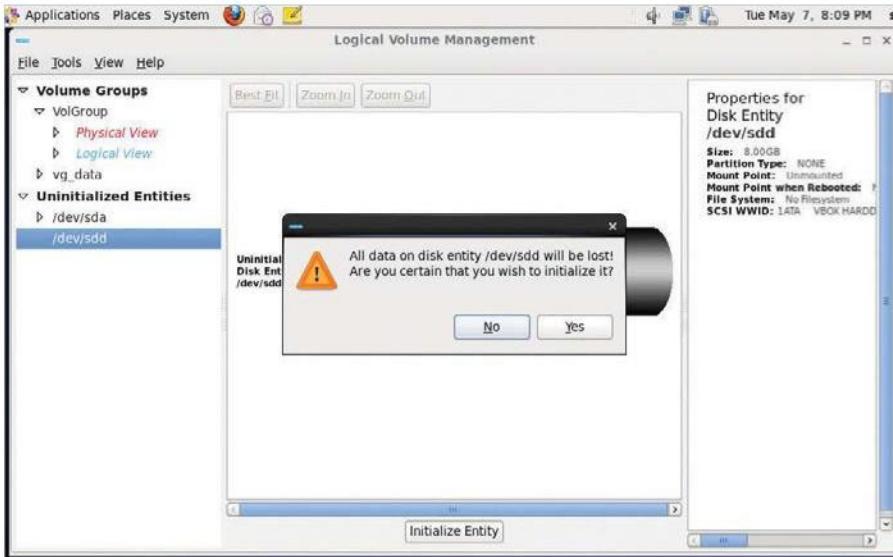
```
[root@localhost ~]# resize2fs /dev/
mapper/vg_data-lvdata
resize2fs 1.41.12 (17-May-2010)
Resizing the filesystem on /dev/mapper/
vg_data-lvdata to 2219008 (4k) blocks.
The filesystem on /dev/mapper/vg_data-
lvdata is now 2219008 blocks long.
```

## 09 Managing LVMs the GUI way

There is actually a GUI tool, `system-config-lvm`. We purposely didn't mention this earlier because you need to understand how all the components of how the system fits together. Not only that but also not all systems, especially servers have a GUI installed. To install the package install `system-config-lvm.noarch`. This will install the graphical LVM tool.

To try it out, let's try with a basic server with an additional 1 GB drive.





## 10 Creating our physical volumes (again!)

Start by loading the LVM tool. If you look to the left-hand side of the application you will see Volume Groups, Unallocated Volumes, and Uninitialized Entities. It shows the volume groups that we work with, volumes and raw devices that are not members of volume groups.

Doing it this way can be very useful in that it will set up the disk label and such automatically, so no terminal and fdisk are required.

Before you add your unused partition, you will need to initialise the partition, as you can see. Do this by clicking Initialize Entity. This is equivalent to preparing a volume using fdisk and pvcreate as we did before.

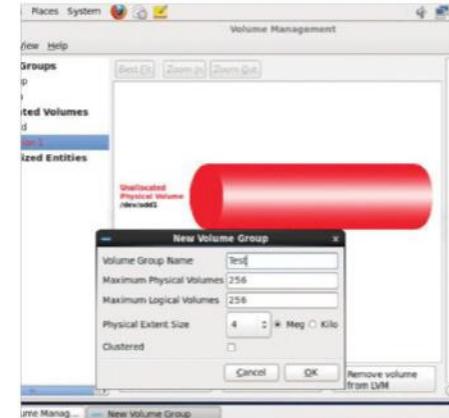
## 11 ...And the volume group

Once this has completed, you will see that the disk has moved from Uninitialized to Unallocated – so, basically, it is prepped to add to our volume group.

You will also notice that if you click on the unallocated volumes you will get a group of buttons below, each stating its purpose.

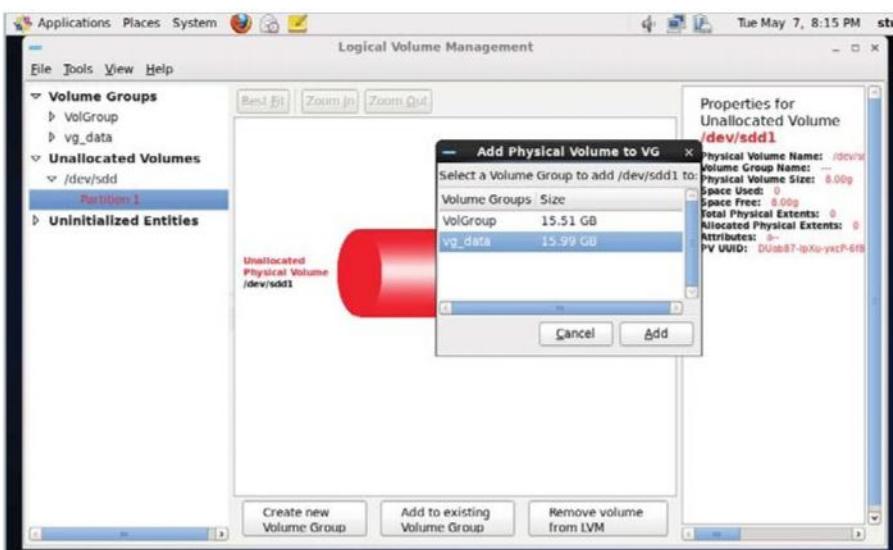
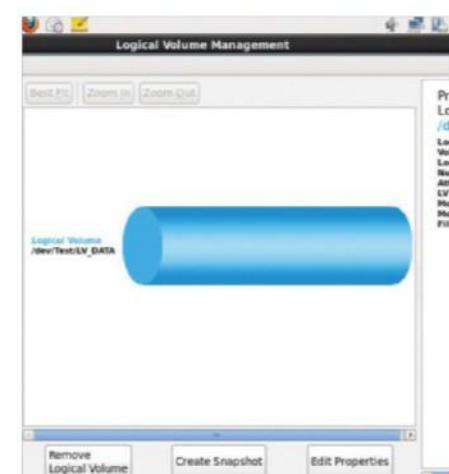
You could, if you wanted, create a new volume group with this disk – or, as we did before, add the disk to an existing volume. Remember, a VG is just a chunk of space that you can split up as you wish; so when you are presented with the sizing screen, carve up that space as you wish.

When creating a new volume group, you will get a selection, as shown below.



## 12 Creating a new volume group

You can leave these options as they are as they're reasonable defaults. At this point you will see the make-up of your new volume group, the physical disks and the logical make-up.



## 13 Carving out storage from our LVM

The last stage is to carve up a chunk of space, so select the Logical view for the VG you want to carve data out of. Hit the Create Logical Volume button. Now you can specify the logical disks.

So, it's easy enough. Give it a good name. Again for LV properties, just go with the defaults. Size is easy enough and obvious enough. If you want to use the entire disk (most people do), click Use Remaining.

Now we have set the size, we need to decide what file system to use. As we are using CentOS 6, our default file system is ext4. Unless you have a compelling reason to change to it, leave the default selection as it is.

All file systems need a mount point to access them, so let's put in the mount points.

**1 Import libraries**

These are the libraries we are going to be using for this program

**2 Set up variables**

These are some variables we'll use to keep track of the script's progress

**3 Initialisation**

This is the initialising function that we will use to handle the input from the user

```

01 import os, sys, urllib2, argparse, datetime, atexit
from bs4 import BeautifulSoup

addresses = []
deepestAddresses = []

maxLevel = 1
storeFolder = "Wikistore " + str(datetime.datetime.now().strftime("%Y-%m-%d %H:%M"))

02 undesirables = [ {"element" : "table", "attr" : {"class' : 'infobox'} }, {"element" : "table", "attr" : {"class' : 'vertical-navbox'}} , {"element" : "span", "attr" : {"class' : 'mw-editsection'}} , {"element" : "div", "attr" : {"class' : 'thumb'}} , {"element" : "sup", "attr" : {"class' : 'reference'}} , {"element" : "div", "attr" : {"class' : 'reflist'}} , {"element" : "table", "attr" : {"class' : 'nowraplinks'}} , {"element" : "table", "attr" : {"class' : 'ambox-Refimprove'}} , {"element" : "img", "attr" : {"None"}, {"element" : "script", "attr" : {"None"}, {"element" : "table", "attr" : {"class' : 'mbox-small'}} , {"element" : "span", "attr" : {"id" : "coordinates"}}, {"element" : "table", "attr" : {"class' : 'ambox-Orphan'}} , {"element" : "div", "attr" : {"class" : "mainarticle"}}, {"element" : {"id" : "References"} } ]

03 def init():
    parser = argparse.ArgumentParser(description='Handle the starting page and number of levels we\'re going to scrape')
    parser.add_argument('-URL', dest='link', action='store', help='The Wikipedia page from which we will start scraping')
    parser.add_argument('-levels', dest="levels", action='store', help='How many levels deep should the scraping go')
    args = parser.parse_args()

    if(args.levels != None):
        global maxLevel1
        maxLevel = int(args.levels)

    if(args.link == None):
        print("You need to pass a link with the -URL flag")
        sys.exit(0)
    else:
        if not os.path.exists(storeFolder):
            os.makedirs(storeFolder)

    grabPage(args.link, 0, args.link.split("/wiki/")[1].strip().replace("_", " "))

    atexit.register(cleanUp)

def isValidLink(link):

```

**Full code listing**

# Scrape Wikipedia with Beautiful Soup

Use the Beautiful Soup Python library to parse Wikipedia's HTML and store it for offline reading

## Resources

### Beautiful Soup:

[www.crummy.com/software/BeautifulSoup/](http://www.crummy.com/software/BeautifulSoup/)

### HTML5Lib:

<https://github.com/html5lib/html5lib-python>

### Python 2.6+ & WikiParser.zip

Six: <https://pypi.python.org/pypi/six/>

In this tutorial we'll use the popular Python library Beautiful Soup to scrape Wikipedia for links to articles and then save those pages for offline reading. This is ideal for when travelling or in a location with a poor internet connection.

The plan is simple: using Beautiful Soup with the HTML5Lib Parser, we're going to load a Wikipedia page, remove all of the GUI and unrelated content, search the content for links to other Wikipedia articles and then, after a tiny bit of modification, write them to a file.

Even though it's now the de facto knowledge base of the world, Wikipedia isn't great when it comes to DOM consistency – that is, IDs and classes are sometimes quite loose in their usage. Because of this, we will also cover how to handle all of the excess bits and bobs of the Wikipedia GUI that we don't need, as well as the various erroneous links that won't be of much use to us. You can find the CSS stylings sheet and a Python script pertaining to this tutorial at <http://bit.ly/19MibBv>.

#### 4 Get the page

Here we grab the page we want to store and remove the bits of the document we don't need

#### 5 Check links

Then we iterate through all of the  tags and check if there's a valid link to another page we can grab, and tweak them for our own use

#### 6 Copy to file

After that, We take the content we've parsed and put it into a brand new HTML file

#### 7 Clean up

Once every page has been parsed and stored, we'll go on through and try to remove any dead links

#### 8 Initialise

This is how we will initialise our script

```

        if "/wiki/" in link and ":" not in link and "http://" not in link and "wikibooks" not in link and "#" not in link and "wikiquote" not in link and "Wiktionary" not in link and "Wikiversity" not in link and "Wikivoyage" not in link and "Wikisource" not in link and "Wikinews" not in link and "Wikidata" not in link:
            return True
        else:
            return False

def grabPage(URL, level, name):
    opener = urllib2.build_opener()
    opener.addheaders = [('User-agent', 'Mozilla/5.0')]
    req = opener.open(URL)

    page = req.read()

    req.close()

    soup = BeautifulSoup(page, "html5lib", from_encoding="UTF-8")
    content = soup.find(id="mw-content-text")

    if hasattr(content, 'find_all'):

        global undesirables

        for notWanted in undesirables:
            removal = content.find_all(notWanted['element'], notWanted['attr'])
            if len(removal) > 0:
                for el in removal:
                    el.extract()

        also = content.find(id="See_also")

        if(also != None):
            also.extract()
            tail = also.find_all_next()
            if(len(tail) > 0):
                for element in tail:
                    element.extract()

        for link in content.find_all('a'):
            href = link["href"]
            if isValidLink(href):
                if level < maxLevel:
                    stored = False
                    for addr in addresses:
                        if addr == link.get("href"):
                            stored = True
                            break
                    if(stored == False):
                        title = link.get('href').replace("/wiki/", "")
                        addresses.append(str(title + ".html"))
                grabPage("http://en.wikipedia.org" + link.get('href'), level + 1, title)
                print title
                link["href"] = link["href"].replace("//wiki/", "") + ".html"

```

04

```

fileName = str(name)

if level == maxLevel:
    deepestAddresses.append(fileName.replace('/', '_') + ".html")

doctype = "<!DOCTYPE html>" + fileName + "</title></head>" + "<body><h1>" + fileName + "</h1>" + str(content) + "</body></html>"

f = open(storeFolder + "/" + fileName.replace('/', '_') + ".html", 'w')
f.write(doctype + "<html lang=\"en\">" + head + "<body><h1>" + fileName + "</h1>" + str(content) + "</body></html>")
f.close()

def cleanUp():

    print("\nRemoving links to pages that have not been saved\n")

    for deepPage in deepestAddresses:
        rF = open(storeFolder + "/" + deepPage, 'r')

        deepSoup = BeautifulSoup(rF.read(), "html5lib", from_encoding="UTF-8")

        for deepLinks in deepSoup.find_all('a'):
            link = deepLinks.get("href")

            pageStored = False

            for addr in addresses:
                if addr == link:
                    pageStored = True

            if pageStored == False:
                if link is not None:
                    if '#' not in link:
                        del deepLinks['href']
                    elif '#' in link and len(link.split('#')) > 1 or ':' in link:
                        del deepLinks['href']

        wF = open(storeFolder + "/" + deepPage, 'w')
        wF.write(str(deepSoup))
        wF.close()

    print("Complete")

if __name__ == "__main__":
    init()

```

05

06

07

08

#### Full code listing continued

"Wikipedia isn't great when it comes to DOM consistency"

```

import os, sys, urllib2, argparse, datetime, atexit
from bs4 import BeautifulSoup

addresses = []
deepestAddresses = []

maxLevel = 1
storeFolder = "Wikistore " + str(datetime.datetime.now().strftime("%Y-%m-%d %H:%M"))

02 undesirables = [ {"element": "table", "attr": {"class": "infobox"}}, {"element": "table", "attr": {"class": "vertical-navbox"}}, {"element": "span", "attr": {"class": "mw-editsection"}}, {"element": "div", "attr": {"class": "thumb"}}, {"element": "sup", "attr": {"class": "reference"}}, {"element": "div", "attr": {"class": "reflist"}}, {"element": "table", "attr": {"class": "nowraplinks"}}, {"element": "table", "attr": {"class": "ambox-Refimprove"}}, {"element": "img", "attr": {"None"}, {"element": "script", "attr": {"None"}}, {"element": "table", "attr": {"class": "mbox-small"}}, {"element": "span", "attr": {"id": "coordinates"}}, {"element": "table", "attr": {"class": "ambox-Orphan"}}, {"element": "div", "attr": {"class": "mainarticle"}}, {"element": "None", "attr": {"id": "References"}} ]

def init():
    parser = argparse.ArgumentParser(description='Handle the starting page and number of levels we\'re going to scrape')
    parser.add_argument('-URL', dest='link', action='store', help='The Wikipedia page from which we will start scraping')
    parser.add_argument('-levels', dest='levels', action='store', help='How many levels deep should the scraping go')
    args = parser.parse_args()

    if(args.levels != None):
        global maxlevel
        maxlevel = int(args.levels)

03 if(args.link == None):
    print("You need to pass a link with the -URL flag")
    sys.exit(0)
else:
    if not os.path.exists(storeFolder):
        os.makedirs(storeFolder)

    grabPage(args.link, 0, args.link.split("/wiki/")[1].strip().replace("_", " "))

    atexit.register(cleanUp)

def isValidLink(link):
    if "/wiki/" in link and ":" not in link and "http://" not in link and "wikibooks" not in link and "#" not in link and "wikiquote" not in link and "wiktionary" not in link and "wikiversity" not in link and "wikivoyage" not in link and "wikisource" not in link and "wikinews" not in link and "wikiversity" not in link and "wikidata" not in link:
        return True
    else:
        return False

```

### WIKI-EVERYTHING

Wikipedia has so many different services that interlink with each other; however, we don't want to grab those pages, so we've got quite a lengthy conditional statement to stop that. It's pretty good at making sure we only get links from Wikipedia.

### INFINITE LINKS

Wikipedia has a lot of links and when you start following links to links to links, the number of pages you have to parse can grow exponentially, depending on the subject matter. By passing through the levels value, we put a cap on the amount of pages we can grab— although the number of files stored can still vary greatly. Use it wisely.

## 01 Install Beautiful Soup & HTML5Lib

Before we can start writing code, we need to install the libraries we'll be using for the program (Beautiful Soup, HTML5Lib, Six). The installation process is fairly standard: grab the libraries from their respective links, then unzip them. In the terminal, enter the unzipped directory and run `python setup.py install` for each library. They will now be ready for use.

## 02 Creating some useful variables

These variables will keep track of the links we've accessed while the script has been running: addresses is a list containing every link we've accessed; deepestAddresses are the links of the pages that were the furthest down the link tree from our starting point; storeFolder is where we will save the HTML files we create and maxLevel is the maximum depth that we can follow the links to from our starting page.

## 03 Handling the user's input

In the first few lines of this function, we're just creating a helper statement. Afterwards, we're parsing any arguments passed into the program on its execution and looking for a -URL flag and a -levels flag. The -levels flag is optional as we already have a preset depth that we'll follow the links to, but we need a link to start from so if the -URL flag is missing, we'll prompt the user and exit. If we have a link, then we quickly check whether or not we have a directory to store files in – which we'll create if we don't – and then we'll fire off the function to get that page. Finally, we register a handler for when the script tries to exit. We'll get to that bit later.

## 04 Retrieving the page from the URL

Here we're using URLLib2 to request the page the user has asked for and then,

once we've received that page, we're going to pass the content through to Beautiful Soup with the `soup` variable. This gives us access to the methods we're going to call as we parse the document.

## 05 Trimming the fat

Wikipedia has a lot of nodes that we don't want to parse. The content variable allows us to straight away ignore most of Wikipedia's GUI, but there are still lots of elements that we don't want to parse. We remedy this by iterating through the list 'undesirables' that we created

**“Wikipedia has a lot of nodes that we don't want to parse”**

```

06   for link in content.find_all('a'):
        href = link["href"]

        if isValidLink(href):
            if level < maxLevel:

                stored = False
                for addr in addresses:
                    if addr == link.get("href"):
                        stored = True

                if(stored == False):
                    title = link.get('href').replace("/wiki/", "")
                    addresses.append(str(title + ".html"))
                    grabPage("http://en.wikipedia.org" + link.get('href'), level + 1, title)
                    print title

                link["href"] = link["href"].replace("/wiki/", "") + ".html"

            fileName = str(name)

            if level == maxLevel:
                deepestAddresses.append(fileName.replace('/', '_') + ".html")

            doctype = "<!DOCTYPE html>"
            head = "<head><meta charset=\"UTF-8\" /><title>" + fileName + "</title></head>"

            f = open(storeFolder + "/" + fileName.replace('/', '_') + ".html", 'w')
            f.write(doctype + "<html lang=\"en\">" + head + "<body><h1>" + fileName + "</h1>" + str(content) + "</body></html>")
            f.close()

```

earlier on in the document. For each different div/section/node that we don't want, we call BeautifulSoup's find\_all() method and use the extract() method to remove that node from the document. At the end of the undesirables loop, most of the content we don't want any more will be gone. We also look for the 'also' element in the Wiki page. Generally, everything after this div is of no use to us. By calling the find\_all\_next() method on the also node, we can get a list of every other element we can remove from that point on.

## 06 Grabbing the links

By calling content.find\_all('a') we get a list of every  in the document. We can iterate through this and check whether or not there is a valid Wikipedia link in the 's href. If the link is a valid link, we quickly check how far down the link tree we are from the original page. If we've reached the maximum depth we can go, we'll store this page and call it quits, otherwise we'll start looking for links that we can grab within it. For every page we request, we append its URL

**“Beautiful Soup is a fast, elegant framework that works with a number of Python HTML parsers”**

to the addresses list; to make sure we don't call the same page twice for each link we find, we check if we've already stored it. If we have, then we'll skip over the rest of the loop, but if we've not then we'll add it to the list of URLs that we've requested and fire off a request. Once that check is done, We then do a quick string replace on that link so that it points to the local directory, not to the subfolder /wiki/ that it's looking for.

## STYLING

Currently, the HTML page will use the built-in browser styles when rendering the page. If you like, you can include the style sheet included in the tutorial resources to make it look a little nicer. To use it, you can minify the script and include it inside a `<style>` tag in the head string on line 102, or you can rewrite the head string to something like:

```
head = "<head><meta
charset=\"UTF-8\"/><title>" +fileName + "</title><style>" + str(open("./PATH/TO/
STYLES", "r").read()) + "</
style></head>"
```

## 07 Writing to file

Now we create a file to store the newly parsed document in for later reading. We change any '/' in the filename to '\_' so the script doesn't try and write to a random folder. We also do a quick check to see how many links we've followed since the first page. If it's the max level, we'll add it to the deepestAddresses list. We'll use this a little bit later.

## 08 Tying up loose ends

After our script has iterated through every link on every page to the maximum level of depth that it can, it will try to exit. On line 34 of the code (on the disc and online) in the init function, we registered the function cleanUp to execute on the program trying to exit; cleanUp's job is to go through the documents that we've downloaded and check that every link we've left in the pages does in fact link to a file that we have available. If it can't match the link in the href to a file in the addresses list, it will remove it. Once we're done, we will have a fully portable chunk of Wikipedia we can take with us.

# Apps

## Great distros and applications

### 136 openSUSE 13.1 RC 1

The next step in the Linux distribution for everyone to use

### 138 Linux Mint 16 RC

The Cinnamon-flavoured revolution is here

### 140 Tails 1.2

How much has Tails matured since its last incarnation?

### 142 Fedora 19 Schrödinger's Cat

How has the latest edition of Fedora shaped up?

### 144 LXLE 14.04

The lightweight distro that aims to keep an aging PC running

### 146 Ubuntu 14.04 LTS

Ubuntu's latest long-term support release with five years of support

### 148 Geany

A fully featured IDE that's a little more lightweight than most

### 149 Eclipse

How does it fare against more community-run efforts?

### 150 wattOS R8

Does the switch from Ubuntu to Debian make this distro better?

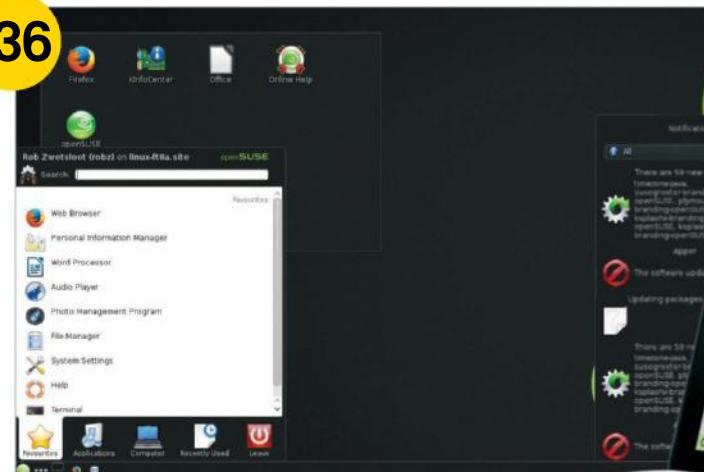
### 152 Dropbox

No introductions necessary for the king of cloud storage

### 153 SpiderOak

A veteran in Linux cloud storage options

136



138





153

**154 OpenShot**

An intuitive yet professional movie-making option

**155 Kdenlive**

A full-featured video editor – it's the complete package

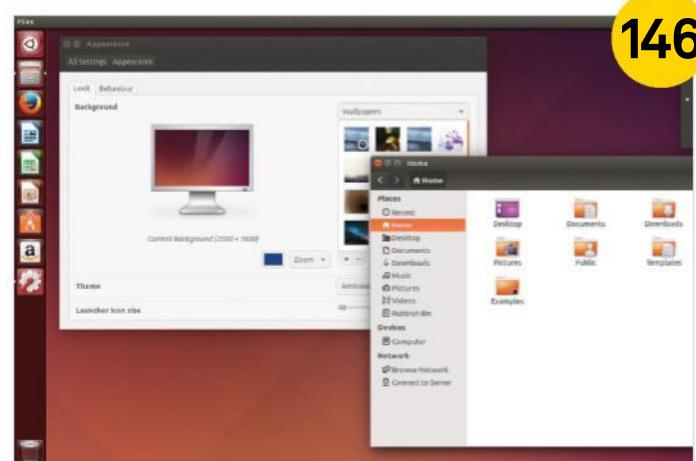
**156 Clementine**

A Linux favourite, how is the latest Clementine player?

**157 Banshee**

Not as popular as Clementine, but still a great option

**“Some of the best free and open-source applications around”**



146

App selection in openSUSE is very good, with a packed selection of default apps and a full repository



KDE and GNOME are still the standard desktop environments, and have been themed and tweaked by the developers

There are a selection of standard updates and bug fixes, but YaST is now completely written in Ruby



# openSUSE 13.1 RC 1

A look at the release candidate for openSUSE's 13, the next step in the Linux distribution for everyone to use

**Pros**  
YaST is now built in Ruby, allowing for easier development, and is still as easy to use as ever

**Cons**  
No live booting off the full DVD, and a limited selection of official desktops compared to some distros

The 13.x line of openSUSE releases is just about here, ready to move beyond the troubled development woes that the community experienced last year for the early releases of 12.x. While there are only a few changes coming to the next version of openSUSE over the previous ones, there are some wide-reaching effects to various levels of users.

One of the most important changes implemented in 13.1 is porting YaST to Ruby. Previously, the openSUSE control centre software was built in its own proprietary language, meaning few people in the community were able to easily contribute to its code. The port to Ruby has been a straight job, and it was introduced as part of

the distro during a beta version of 13.1. For the desktop user, this may not mean much, but to the developers and the community it's a huge step forward in allowing one of the major features of openSUSE to be much more open and friendly to those who want to commit changes.

With the first iteration of such a port let loose in the wild, it's natural to be concerned over the new YaST's stability and quality – thankfully, the porters seem to have done an exceedingly good job. The control centre is as usable as it's ever been, and there were no issues using it for adding and removing software, changing network settings, adjusting the boot menu and all the other tasks it can perform.



■ Installation from the full DVD allows for a complete and customisable install, while the live CDs offer a great preview

## "OpenSUSE is also about community, and the changes to YaST and efforts made with Btrfs are a great indicator of how strong it currently is"

### Minor updates

As well as YaST, there's been some great improvements over Btrfs, the future file system that keeps being just out of reach. It's not a default yet, but the developers and community have been making an effort to improve its support in the latest version of openSUSE with some impressive results. Right now it's considered safe to use, with the intention that it'll be a default in 13.1. We've heard that line before, though, about Btrfs from other developers.

There's a host of updates to all the desktop environments, the Linux kernel has been updated to version 3.11.3 (with the added Btrfs patches) and interestingly, there's an effort to update GStreamer from 0.1.0 to the newer 1.0 – although this hasn't been implemented as of the release candidate.

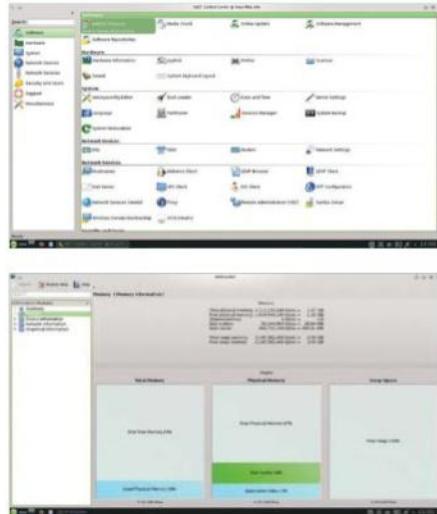
Aside from the big changes to YaST, it's somewhat of a safe update for openSUSE. There's no problem with this, of course, though, and it's allowed it to stay rock solid and compatible with

a lot of hardware types and keep its great user experience intact.

### Rousing performance

Updates and new features aside, openSUSE 13.1 still works as advertised. The images supplied come in three main flavours – two live discs containing one of the two main desktop environments, and the full installation DVD. The KDE and GNOME spins allow you to live-boot into openSUSE and give it a test before committing to installing, while the DVD version is specifically just for installation.

The DVD installer is still one of the better Linux installers out there. The dedicated process is split up into distinct sections with a logical flow to the process. Default options are passable for the lower-end users, while there's plenty of room for customisation and further setup for the more advanced users that encompass the targeted user base of openSUSE itself. You can also choose between the main supported desktops, or select a



■ System information is easily accessible, allowing for system diagnostics on every level

more lightweight alternative if you require it. The only thing really missing is adding or removing different software packs, the kind that the Mageia installer provides. While you can make your own custom ISO that will do this for you with SUSE Studio, it would be nice to have even a basic version of it with the official release.

Installation is quite fast, and will automatically restart and dump you into the desktop. The openSUSE desktop themes continue to be some of the best around, with great aesthetics and design ethos that eke a little bit more out of the standard KDE and GNOME.

### The next generation

Right now, then, everything looks fantastic for the next openSUSE. The philosophy of the distro has always been about making it the best OS to use for novices and veterans alike. This is again accomplished with a fantastic selection of tools for sysadmins to manage the systems locally or remotely, and a smart design that allows normal desktop users to quickly get into a new workflow rhythm. OpenSUSE is also about community, and the changes to YaST and efforts made with Btrfs are a great indicator of how strong it currently is. We look forward to seeing what the next version brings.



The community distro returns with a new version of YaST and a series of updates that still allow it to be one of the most usable and stable distros available. The 13.x line of openSUSE may be the best yet.

Slight changes to the transparency effects and colours make navigating the already great Mint Menu just that bit better



Edge tiling is improved with a new lock feature that means windows you wish to stay visible always will



Cinnamon 2.0 is very stable, even with its own brand new back-end

# Linux Mint 16 RC

The Cinnamon-flavoured revolution is here as Mint releases its first truly independent desktop environment

## Pros

Cinnamon is ever so slightly yet noticeably better, even though it's had more work on the back-end than front-end

## Cons

Mainly a stepping stone to Linux Mint 17, and you still need a more modern system to make the most of Cinnamon

From a normal user-perspective, Linux Mint 16 might be the lightest Mint release in terms of new features and content.

There are some aesthetic changes to Cinnamon, MATE is still roughly the same and there's a new default theme for MDM. It's still the same old, fantastic Linux Mint as before in that regard; however, this seemingly minor release has one of the biggest updates in Linux Mint history. Cinnamon 2.0, the Mint-developed desktop environment, has replaced its GNOME base with one of its own.

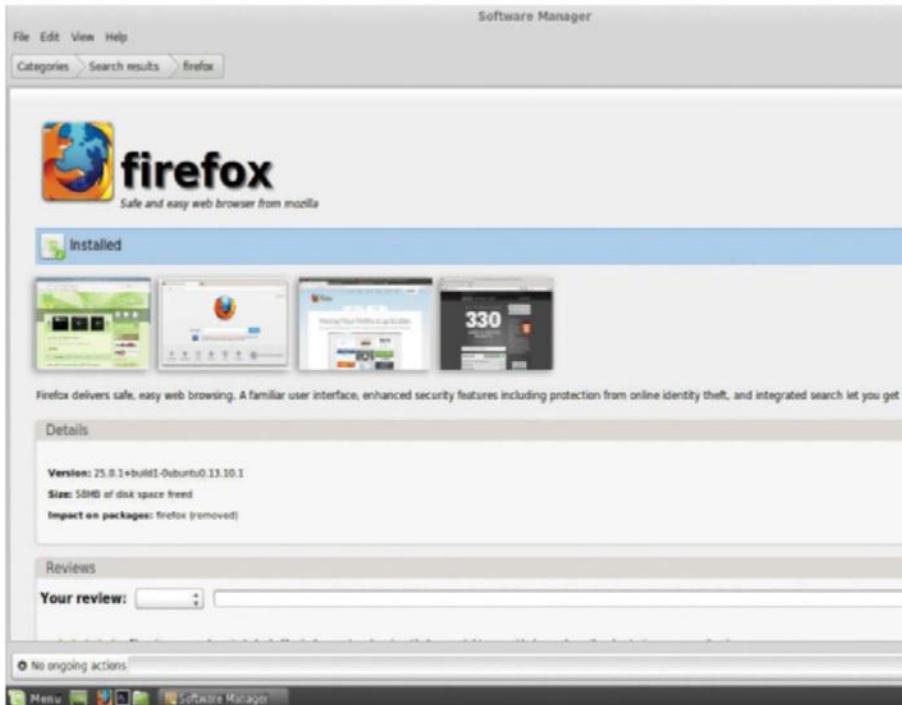
We'll get to that, though – first of all, here are the facts. Linux Mint 16 is based on Ubuntu 13.10, the pre-LTS release, and as usual gets rid of some of the more commercial stuff while keeping the non-free software, making it a better experience for users not too concerned with strictly using FOSS. Cinnamon and MATE versions are offered to start off, with specific Mint-themed KDE and Xfce versions to come.

These desktops are already available through the software repositories, though.

The installation hasn't changed at all since last time, using the same standard Ubuntu installer. It's fast, fairly easy to use and has some decent default options for installing alongside or replacing a distro altogether. It's themed to Mint at least, so there's no mistaking what you're installing.

## Cinnamon Spiced

Once installation is completed, though, you restart into one of the first new changes – the new HTML 5 login theme. It's a nice little touch, and it looks little like the Ubuntu login screen. This new MDM theme also has your user selected by default, rather than the previous method of having to manually enter a username or select a user. The previous clouds theme is still available, nonetheless, and you can also switch to a more



■ The Software Manager is one of the things that separates it from Ubuntu, with no ads or paid apps. It's had a few minor updates, including the ability to show off more screenshots

## "We couldn't have asked much more of Linux Mint for this release"

secure login screen if the idea of having your username on display worries you.

Here's where the most interesting part starts – Cinnamon 2.0. On first impressions, long-term Cinnamon users will notice that the icons and Mint Menu are just ever so slightly different: brighter colours and better transparency effects, although the fonts look a touch fuzzier. Generally though, it just highlights the different important areas a bit better, slightly aiding in navigation and workflow, especially for new users. While these are only little touches, and there are a more of them scattered throughout, the biggest change is the aforementioned change to a pure Cinnamon base.

While invisible to most, the removal of the GNOME back-end from the Cinnamon code is an enormous achievement for the Mint team, especially as the final result is very stable. No functionality has been lost in the transition and while nothing has been specifically added either, it means that future versions of Cinnamon will be able to include more innovation than before.

Such as new feature edge-snapping, allowing you to lock a window to a specific corner or side of the desktop and other windows will maximise around it. This is great for multi-monitor setups and/or extremely large screens, and lets you keep an eye on windows that are currently important.

### Our other MATE

While Cinnamon is steaming ahead with big changes, the same thing can't be said for MATE in this release. 1.6 from April is still in use, which is still a great desktop environment but it's being slightly shown up by the rest of the distro. The team has been concentrating more on getting it stable on other distros, which has really been the main issue with MATE since its inception. While it's doing that well, it means it's not receiving much innovation for the time-being.

This is sort of indicative of Linux Mint 16 in general. When we spoke to Clem Lefebvre around the release of Linux Mint 15, he mentioned that 16 would be a 'harder sell'



■ MATE is still at 1.6, but it's a great desktop and a solid choice for a Mint install nonetheless



■ MDM has come on in leaps and bounds since its first introduction, and this release is the first time it's seemed truly modern

as it wasn't receiving the wealth of great new features as 15 was. However, it's important that they release it like this now so that when Ubuntu 14.04 LTS is released, the crowdsourced bug testing of a major release like this will iron out any minor issues still remaining.

The big question, though: is this still the fantastic Linux Mint distro we've come to expect? The answer is a resounding yes – while very little is new, with such a big change it's much more important that they were able to at least replicate Linux Mint 15. They've managed to do that and a little more, with the result being a great Linux distro that, while great for a home office, can be used anywhere.

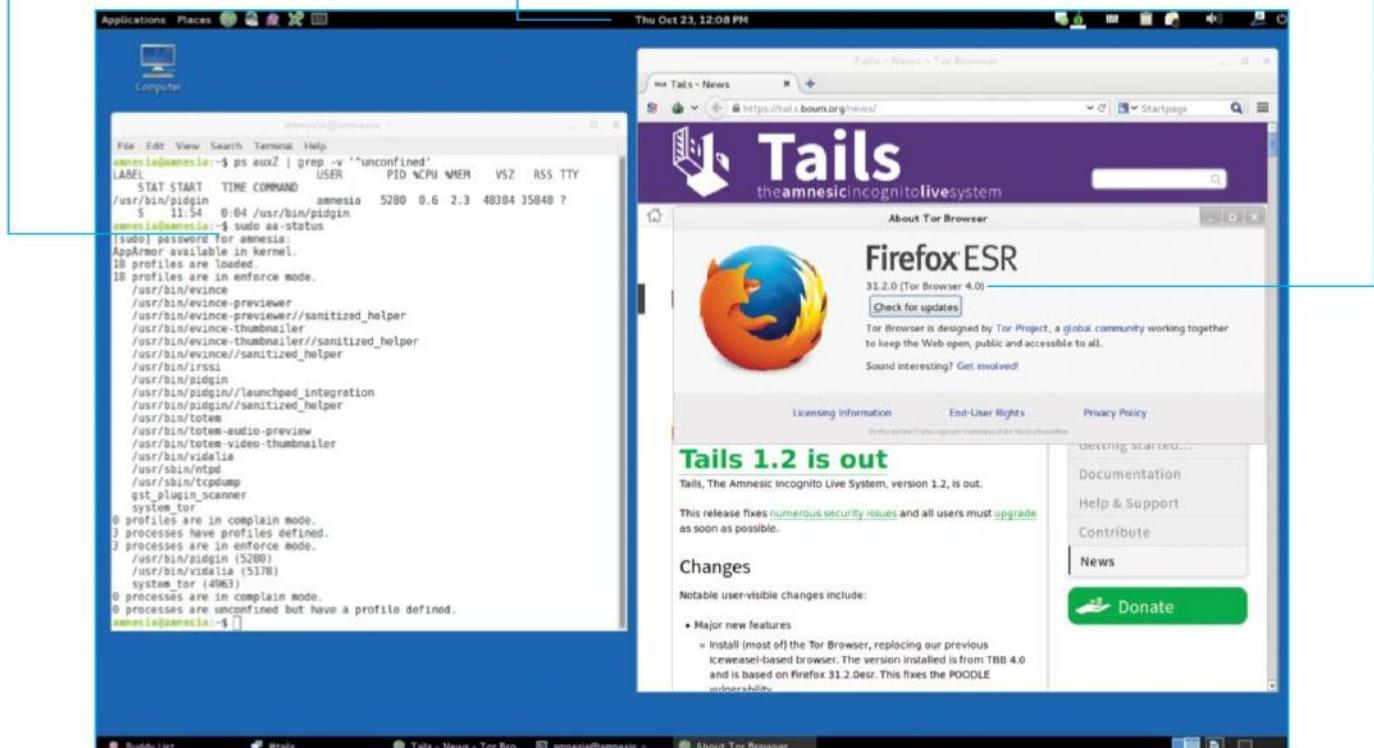


We couldn't have asked much more of Linux Mint for this release, but it's managed to meet all our expectations and slightly exceed them. An important release and frankly still a great distro, politics or otherwise.

AppArmor is a new feature that restricts which system resources running applications can access

The GNOME 3 desktop environment has been introduced since we last reviewed this distro in issue 140

Iceweasel has been removed from Tails and replaced with a modified version of Tor Browser 4.0



# Tails 1.2

Beefed up with the latest Tor Browser, AppArmor and a slew of package updates, how much has Tails matured since we last looked at it?

## Pros

Near watertight security. Works perfectly out of the box and packed with advanced options too.

## Cons

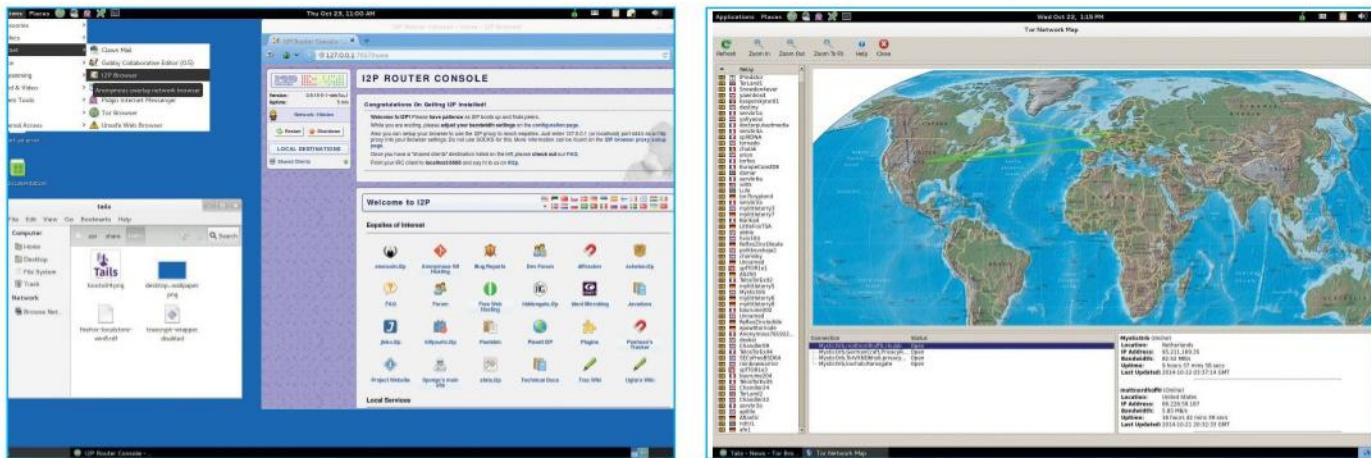
Tor Browser 4.0 is only available in 15 languages so far and there's still a substantial list of known issues

**In the short time since it first emerged, Tails (The Amnesic Incognito Live System) has become an indispensable tool for those needing to protect their identity online, whether they are journalists, activists, citizens with censored internet connections or those concerned with reduced net neutrality and increased state snooping.** It can be cloned onto a disc, USB stick or SD card, lives entirely inside your RAM to avoid leaving traces on your hard disk and configures Tor on boot so that you can immediately begin interacting anonymously online.

Tails has had some pretty sweeping changes made to it since we first took a look at version 1.0 back in May. The relatively recent Tails 1.1 update focused more on the day-to-day distro usability, replacing OpenOffice with LibreOffice and updating thousands

of packages with the move across to a Debian Wheezy base, as well as introducing the GNOME 3 desktop. The Windows XP camouflage mode was also updated to Windows 8, but is still only decent enough to satisfy a passing glance. Other notable changes were the reintroduction of VirtualBox guest modules, though only for 32-bit kernels, and the implementation of UEFI boot support for more modern machines. Regarding the former, guest additions are now enabled by default in Tails 1.2, which means your display should now automatically be configured on boot if you're virtualising.

Version 1.2 is very much concerned with your privacy and security. The biggest change with this release is that the custom browser, which is based on Firefox and Iceweasel patches, has



■ Enable I2P through the boot menu and you'll be able to use the I2P Browser to access the anonymous overlay network

■ You can check the status of open connections through the Tor Network Map and become a relay yourself via the Tor Settings

## "Guest additions are now enabled by default in Tails 1.2, which means your display should now automatically be configured on boot"

been completely swept out and replaced with Tor Browser 4.0. Based on the Firefox 31 extended support release, this version of Tor benefits from Firefox's recent security updates and has also had SSLv3 disabled in order to protect users against POODLE attacks. There's also a new updater for the browser, accessible through the menu's Help button: click About Tor Browser and you'll see your new button. Three versions of the meek pluggable transport have also been added to Tor Browser 4.0, designed specifically for people using a censored internet connection. Meek uses domain fronting to send a message to a Tor relay that's very hard to block, since different domain names are used at different communication layers – essentially, the censor will believe you are accessing Google App Engine, Amazon CloudFront or Microsoft Azure, depending on how you've configured meek through the Tor Browser Bundle installer; after answering Yes when asked if your connections to the Tor network are being censored, you simply choose one of the meek transport types from the 'Connect with provided bridges' drop-down. Unfortunately, this option doesn't seem to be present in the Tor Browser shipping with Tails 1.2 or through the additional login options, but the release notes say that "most of" Tor Browser has been installed so this is perhaps an upcoming feature in Tails – keep an eye out.

I2P was also upgraded to version 0.9.15, which fixes a security hole from the versions used in Tails 1.1 and earlier. I2P is another anonymising network included in the distro since Tails 0.7. It's no

longer enabled by default, in order to reduce any attention you might garner by running two anonymising networks at once and also to reduce the CPU overhead. You can change that by adding the i2p option to the boot menu, undoing the work of one of the scripts that moves I2P into an 'I2P-disabled' folder and adding the I2P Browser to your Applications menu, through which you can visit eepsites. TrueCrypt is disabled by default and will be removed in Tails 1.2.1, but the Tails documentation shows you how to open TrueCrypt volumes using cryptsetup.

Another important change is that AppArmor is now being used to isolate applications. Currently it is being used to ringfence resources on the file system, using a number of security policies to define which resources can be accessed by applications and with which privileges. You can check on the status of AppArmor profiles by using these two commands: `ps auxZ | grep v` "unconfined" and `sudo aa-status`. The developers behind its introduction to Tails are part of the Debian AppArmor team and are (at the time of writing) working to update it for Debian Jessie, so that Jessie will contain everything that's needed in Tails 1.2.1 before its freeze date.

There's still a lot of work to be done on the already formidable Tails and, from what we can tell, it's going to become mighty indeed. Version 2.0 will focus on "sustainability and maintainability" and 3.0 will focus on "hardening and security". At version 1.2, we're already impressed with the fantastic work being done by the Tails community to help protect our privacy.

### Summary

Designed to be used with little-to-no setup, Tails takes care of your security and anonymity quickly and quietly, providing a great suite of apps, including Poedit, Gobby, Scribus and more, alongside essentials like KeePassX and GnuPG. Version 1.2 improves an already fantastic distribution.



**Download now**

[tails.boum.org](http://tails.boum.org)

GNOME is not for everyone; luckily, Fedora offers more options as well



Settings are presented separately in the search results, making for quick access to some options

The search function is great for finding apps and documents alike

# Fedora 19 Schrödinger's Cat

After a long-delayed and divisive Fedora 18, how has the latest edition of Fedora shaped up?

**The famous Schrödinger's cat experiment is one of those stories from history that is perceived incorrectly in popular culture – like King Cnut arrogantly trying to stop the tide, or Bill Gates saying that 640K would be enough for everyone.** Erwin Schrödinger's hypothetical experiment was actually a way of explaining how some interpretations of quantum mechanics were a contradiction of common sense. While this name was voted on for Fedora 19 by the masses of the internet, it's sort of indicative of the kind of problems people have been having with the default state of the distro for the last few iterations. GNOME has been moving quickly away from the traditional desktop metaphor

for years, with recent updates going against a mouse and keyboard workflow. The Anaconda installer update from Fedora 18 limited some options in favour of a more aesthetically pleasing experience. The distro has also not been particularly bug free, with systemd causing headaches for some. So, with Fedora 19 have some of these immediate issues been addressed, or are there new ones to throw on the list?

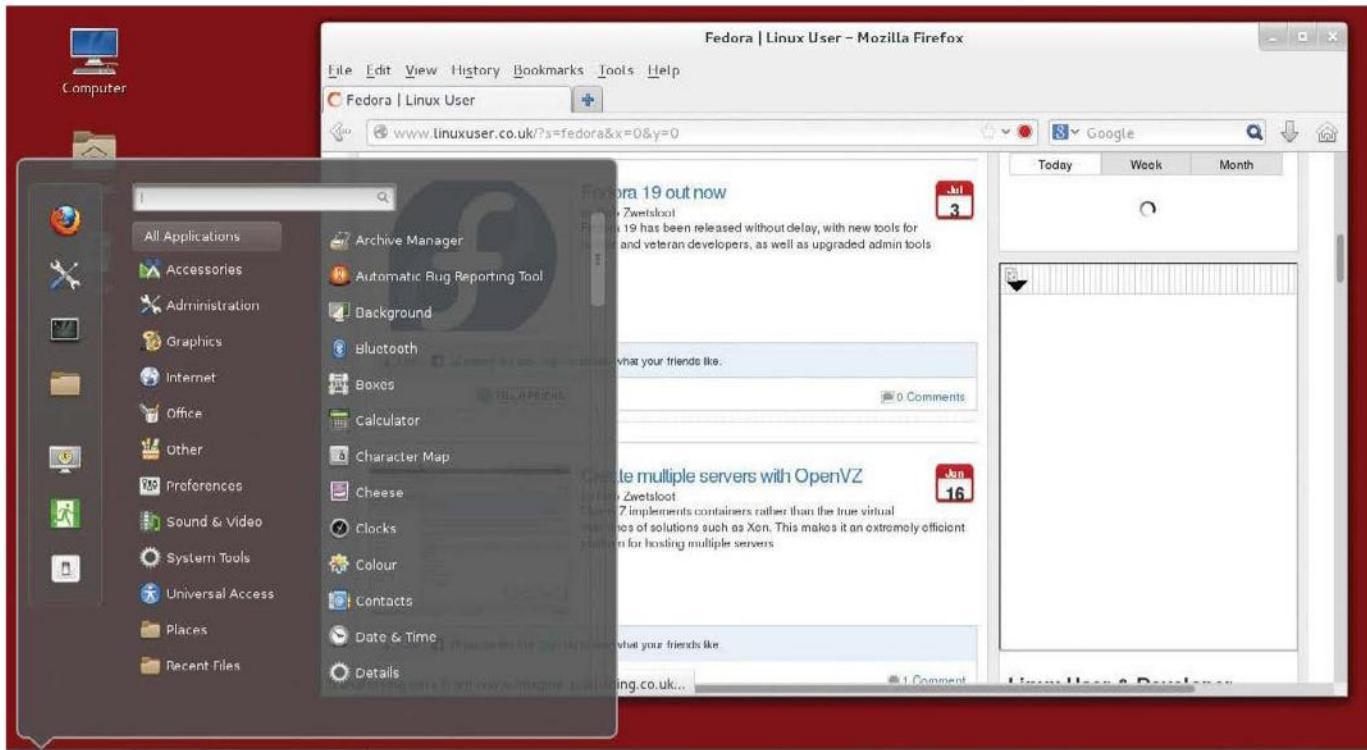
## Upgraded installer

The first thing you'll experience with Fedora 19 is the installer, which has been upgraded again. Hardware recognition seems fine and there's

**Pros**  
The new installer is a bit better than previous ones and the new 3D printing tools are a nice addition

**Cons**  
Still tripping over some of the more controversial, cutting-edge features that have caused issues

now a lot more control over the partitioning and editing of storage locations, an issue a lot of people had with Fedora 18. However, the method of doing so is not the most straightforward. Like in other graphical installers, you can select the hard drive you wish to use; however, instead of then performing a manual partition, or selecting a recommended installation scheme, you need to start 'reclaiming' space. This can be done by either completely deleting any existing partitions, resizing, or creating your own through the reclaim option, otherwise it will automatically try to fill the space already made. Pre-existing swap partitions are ignored, for some reason, and 19 will create its own if space



## "There's now a lot more control over the partitioning and editing of storage locations"

is cleared out. The installation will start before you can finish creating a root password or user, saving some time, but it still seems that this new installer is not ready and needs a lot more time in the oven.

### **GNOME 3.8**

If perhaps the installer is supposed to be more in line with the simplification of GNOME, it's doing a good job. GNOME 3.8 hasn't had many major changes over 3.6, insomuch that it's still 'dumbed down' in many respects. As if to highlight that this is the path the GNOME project is taking, a video explaining how to use GNOME launches on first boot. Credit where it's due, though – the search function launched from whatever the Windows key is being called these days has always been good. Even if it's

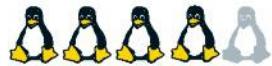
supposed to be a substitute for a large amount of a workflow, the search function part is often faster than mousing around and has now been upgraded to include some system settings results in your search. Sort of like a hybrid between the same functionality in Unity's HUD and the classic search results, but without an unnecessary split between them, or the inclusion of Amazon adverts.

### **Stay up to date**

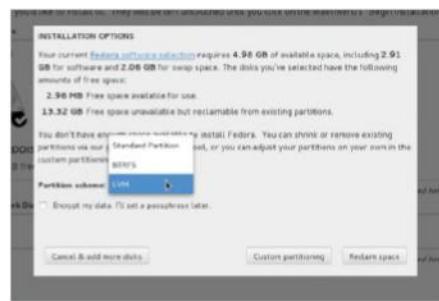
The software updater has also been separated from the generically termed 'Software' package manager now, as well as in the applications list, although it's still accessible from there. It's here and in the repos that you can access all the alternative desktops, although there are three extra spins of Fedora that you can also use. As well as the KDE one, there's the lightweight Xfce and LXDE choices, with other popular desktops such as Cinnamon 1.8 and MATE available in the repos. This version of Cinnamon is built to work on GNOME 3.8, so you won't need to downgrade.

The distro itself is more stable than Fedora 18 on our physical setup. In a virtual machine, though, we experienced some noticeable slowdown and minor graphical glitches – so for virtual distribution you may need to do testing before deployment. Fedora, then, is not quite

the beast it used to be, with its cutting-edge stance harming it more than it has in the past. For those that were using Fedora 18 without issues, it's a great upgrade; however, for those that moved away in recent years, this won't bring you back. The box contains only one quantum waveform – and it's not looking good for the cat.

 The latest version of Fedora has fixed some of the problems we experienced with the previous editions. However, there's still a way to go for some of its features.

■ The installer has been a sore spot for Fedora users since 18 and while the new one takes some steps to fix the issues, it still needs development



Access ROXTerm with a quick Alt+X, or open a directory into the terminal right from PCManFM

Shade windows to reduce them to their title bars and reduce desktop clutter without minimising as you switch tasks

You can switch between two different desktops and also pin windows so that they show up on both



# LXLE 14.04

LXLE updates for the Lubuntu 14.04 LTS release – but can this lightweight distro really keep an aging PC going for two years?

## Pros

Very nippy, light on resources and packs an impressive software loadout on installation

## Cons

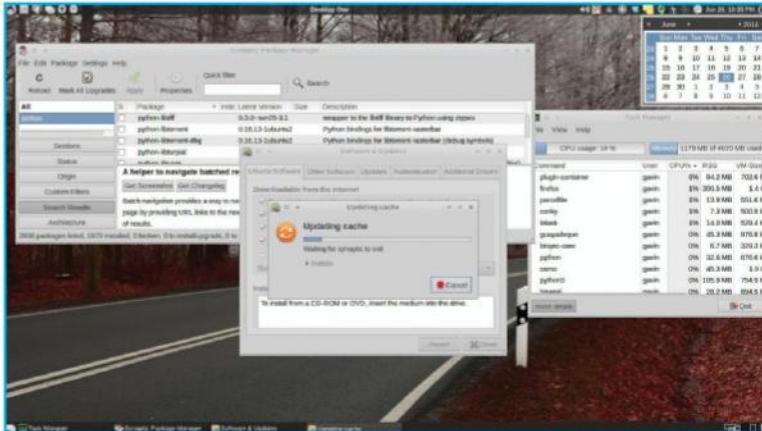
Seasoned users might want a little less app-bloat on boot in order to reduce the storage space it needs

**LXLE has been kicking around for a while now and, for a supposedly lightweight distro, it's looking fearsomely feature-packed right now.** Having said that, it's hard not to love LXLE, as it's treading the line between resource efficiency and usability pretty well, and is borderline addictive when it comes to the DE itself. The clue's in the updated acronym; rather than standing for 'Lubuntu eXtra Life Extension', as it did in the days before Lubuntu LTS releases, when LXLE was around to fill that niche using the LXDE desktop environment, it's now pitched as the 'LXDE eXtra Luxury Edition'.

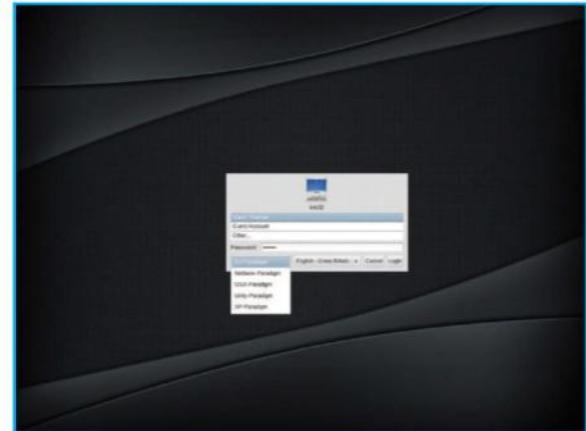
Based on the LUbuntu 14.04 long-term support release, LXLE is for users of 'aging PCs' and, reading between the lines, is aiming itself squarely at Windows XP users. The installation process is simple and straightforward, taking just a few minutes and requiring no manual partitioning and not really needing any

follow-up configuration. Once you hit the login screen, you can choose from five desktop paradigms: GNOME 2, Unity, Windows XP, Mac OS X and Netbook, with the latter providing the most bare-bones environment for small-screen Netbooks with low system resources. While the other four paradigms are only relatively minor cosmetic adjustments, changing the positions of the dock and menu bars and occasionally conflating things to match the paradigm's target look, the real advantage here is that they will be useful for all switchers. So if you're a seasoned Linux user who's a fan of lower, centralised docks or a refugee from the abandoned Windows XP camp, you can change your framework to suit and speed up acclimatisation.

But the real core of the design is pure LXDE. There's a neatness to the whole thing that's down to the font choices and the icon and panel designs as much as the menu structures.



■ There are plenty of options for updating software and managing packages, including the main Ubuntu and Lubuntu centres



■ You can choose between the GNOME 2, OS X, XP, Unity and Netbook desktop paradigms at the login screen

## “LXLE uses ROXTerm as its main terminal emulator which is plenty customisable and quick-launches with a nicely mapped Alt+X”

You can switch between two desktops, with their thumbnails showing outlines of opened windows (it's a nice touch, but it would be nice to be able to select these directly, rather than just select the desktop). There's a button in the corner that lets you minimise all open windows ('iconify') or shrink them to show just their top bars ('shade') with a left- or middle-click, respectively. You'll see an at-a-glance panel of CPU info that can be hidden or popped out, and a nice-looking weather panel you can open with the button beside the date and time.

Random Wallpaper is a little button that sits in your menu bar beside the app and file trees, and on-click it randomly changes the desktop background from its library of a hundred images, many of which are stunning. Our main gripe is with the windows themselves, in that the handles for resizing them need close to 20/20 vision and pixel-perfect mousework in order for you to actually grab them, and there are a few graphical bugs in both the 64- and 32-bit versions – but we can live with that. Throw in PCManFM (also in Lubuntu) and suddenly the whole thing really starts working.

LXLE uses ROXTerm as its main terminal emulator, rather than Lubuntu's XTerm, which is plenty customisable and quick-launches with a nicely mapped Alt+X (the app finder/launcher shortcut is Alt+Z, though with that it's only picking up app names – so typing 'email' won't give you Claws Mail, for example). This is where LXLE starts to make its mark

– compared to Lubuntu, which also runs LXDE, the 'eXtra' differences really are apparent. It's not just that XTerm has been replaced with ROXTerm, but more that the LXLE team seems to have really weighed up exactly what software it wants to use – and this curation is apparent.

You have the Parcellite clipboard manager and BitTorrent Sync sat in the menu bar; TLP power management is already available in your terminal, which is great for laptop users; OpenShot and Audacity are provided for video and audio editing; GIMP, Shotwell and Simple Image Reducer handle images; there's Firefox, Claws Mail, Pidgin, Libre Office and Osmo for productivity; the Ubuntu Update Manager, Lubuntu Software Centre, Synaptic Package Manager and Y PPA Manager (plus Steam) give you a load of ways to get more software and keep it up to date. The apps available on boot belies the intentions of LXLE; this LTS release wants to get you up and running right away, with no need to go looking for more FOSS – chances are there's something there already. And it shows in the sizes – the 64-bit ISO is a gig and the distro itself consumes a cool 7 GB.

Thing is though, LXLE is lightweight where it counts. The 32-bit version we tried (an updated 12.04.4) on an old PC was nippy as hell, and it was just blistering with the 64-bit on a more modern machine. It'll certainly breathe new life into aging machines. That, plus the 'ready to go' philosophy of this good-looking LTS distro, has really made LXLE grow on us.

### Summary

It's quick to install, quick to run and boasts a huge array of baked-in software. It might be too much for the more selective software user, but casual users will love this – and LXLE looks gorgeous.



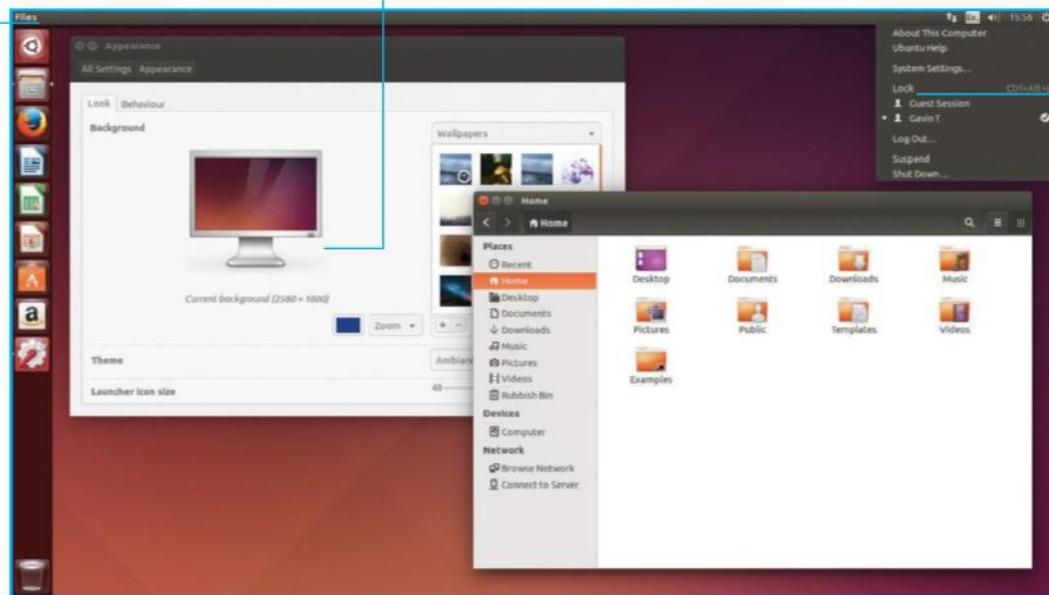
**Download now**

<http://lxle.net>

The menus of the application you're currently using are now available by default in the integrated menubar

Ubuntu's Unity desktop environment can now scale on a per-monitor basis for a better UX with high-DPI screens

Among Unity's improvements are rounded window decorations and a brand new lock screen interface



# Ubuntu 14.04 LTS

'Trusty Tahr' is Ubuntu's latest long-term support release and will be supported by Canonical's development team for the next five years

**Pros**  
A focus on stability  
– par for the course with an LTS release  
– means that you won't need to re-install Ubuntu for five more years, if you keep it updated

**Cons**  
With stability comes a lack of new features, making it an unexciting release for those wanting to be on the cutting edge

An Ubuntu LTS release sets off a long chain of events that goes far beyond Canonical's desktop offering. The different flavours of Ubuntu and the Ubuntu-based distros are just the tip of the iceberg, with LTS-only spins getting a refresh and enterprises possibly getting in on or updating to the newer version. It's an important release for both Canonical and the Linux/FOSS community, and it will mean big business for the former in the coming years.

Some of the promises of new features for Trusty Tahr missed the feature-freeze deadline, though. Canonical's plans for their new display server Mir making it into the LTS have long since been delayed to 2016's 16.04 LTS release, which otherwise leaves the core of the system with relatively few new features.

On the surface, this could very well work in their favour; during the last round of LTS development, Canonical were keen to drive home the message that they were focusing

on the stability of the distro. From a philosophical point of view this makes complete sense – the LTS is the version of Ubuntu people will want to rely on. Focusing on stability will make it better for users, and Canonical will sell more support packages in the process. It benefits everyone involved, as long as the final product actually is stable.

## Trusty release

In the few months since its release Trusty Tahr has certainly lived up to expectations, with a minor point-update resolving early issues almost immediately and a swathe of LTS releases from the likes of Linux Mint, LXLE and other big-name distros all adopting the 14.04 base for the coming years.

As stated before there isn't a massive amount of new core features, however there are some distro-specific changes that can be seen in the selection of official Ubuntu releases. These official ones include Kubuntu, Xubuntu, Lubuntu,



■ The Dash now enables you to search multiple online sources simultaneously, and you can get full-screen previews with a right-click



■ With the Ubuntu Software Centre, you can easily find and install any free or paid-for app that's compatible with 14.04

## "We're interested to see how this is going to shape the future of Ubuntu and other distros"

Edubuntu, Ubuntu GNOME, Ubuntu Kylin and Ubuntu Studio; all the major Ubuntu-spins bar Ubuntu itself.

But there are a few things that are common across all the releases; the Linux kernel included is the 3.13 release, and all the packages have been updated to the latest version. Shortly after the release of 14.04, Canonical released a follow-up version – 14.04.1 – which fixes a few minor bugs from the initial release and includes the 3.13.0-32.57 Ubuntu Linux Kernel, which is based on the stable 3.13.11 upstream Linux kernel. It has effectively replaced the original 14.04 release and is now the main download.

As for using the latest version of third-party software, this is probably the best way for users to ensure the software is relevant, up to date and stable.

### Unique qualities

Ubuntu and its extended family have also received individual updates in this release, mainly tied to their interfaces.

Unity has received probably the smallest update of the lot. As we mentioned earlier, the original plans to work on converging the mobile and tablet versions were abandoned for 14.04. Instead, the focus on stability has meant there are very few new additions. Notable examples include the option to return menu bars to the windows they belong to, rather than living on the top bar. There's also the ability to search your open applications with text entry, previously a compiz feature.

Xubuntu's new features are a little more major, boasting a brand new lock screen as well as lock screen manager in Light Locker, a new menu customisation system as well as some other settings updates on top of that.

Kubuntu is the release with the biggest changes. Muon – Kubuntu's alternative to the Ubuntu Software Center – has had a bit of an overhaul; the codebase is now a little more robust, while the interface has experienced some minor tweaks to support this. A new KDE Software Development Kit gives users a great way to get started with coding KDE and Qt apps. It provides an IDE to work in and it's a nice, quick way to learn how to use Qt.

There's also a new driver manager and a new touchpad config tool, and some major updates to Gwenview, KDE Telepathy/IM client, localisation support, network manager and more. Kubuntu may well be the version of Ubuntu that has the most to offer this time around, however it's something we're not too worried about.

### Same old same old

As we mentioned before, an LTS release is best when it's stable. This means relying on tried-and-tested software, and maybe having a more boring release for users than, for example, the following 14.10 'Utopic Unicorn' release that focuses on servers and the cloud. Still, Ubuntu 14.04 is looking good, and we are interested to see how this is going to shape the future of Ubuntu and many other distros.

### Summary

Not revolutionary, but it's not supposed to be. Delaying features will only make them better further down the line while also reducing the risks for the LTS. As a beta, it's already showing signs of being one of the better releases in years.



**Download now**

[www.ubuntu.com](http://www.ubuntu.com)

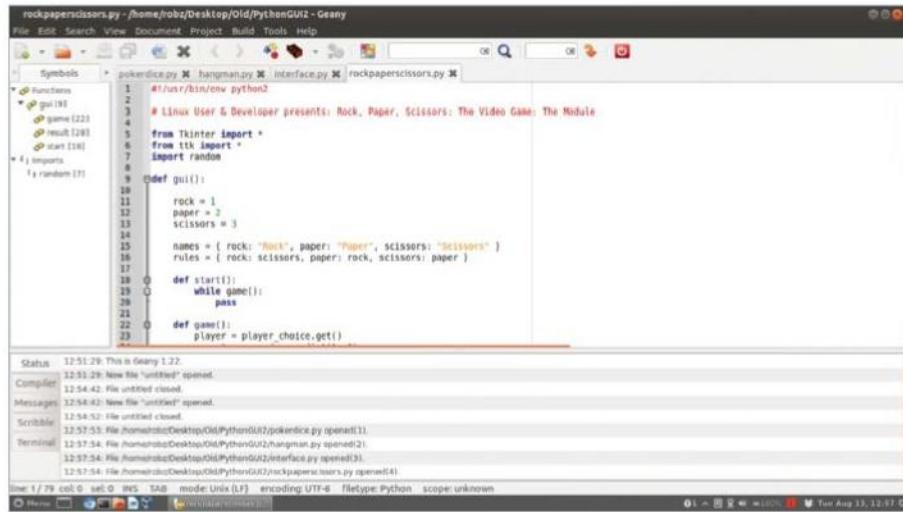
# Geany

A fully featured IDE  
that's a little more  
lightweight than most

**Geany is sometimes mistaken for a glorified text editor.** In reality, it's a full IDE, albeit slightly more lightweight than most.

Geany has multi-language support out of the box, with easy options to create files that use different languages as part of the main interface. Projects are kept in specific folders, like Eclipse, although there is a file to go with them that Geany can read to manage the projects. Multiple projects can be viewed and edited at a time, all in different languages. There are also debuggers and builders for the various languages that support them, and you can run and test stuff like Python that doesn't need compiling as such.

The rest of the interface for Geany is clean and well labelled. Code is automatically highlighted with the correct syntax and there's a smart tab in the left column that allows you to track and view the different variables, functions and classes in the projects and code. Navigation through the code is fairly simple via context-sensitive menus and nice options within the rest of the interface, and the usual code-editing tools such as commenting out a selection or indenting are all present.



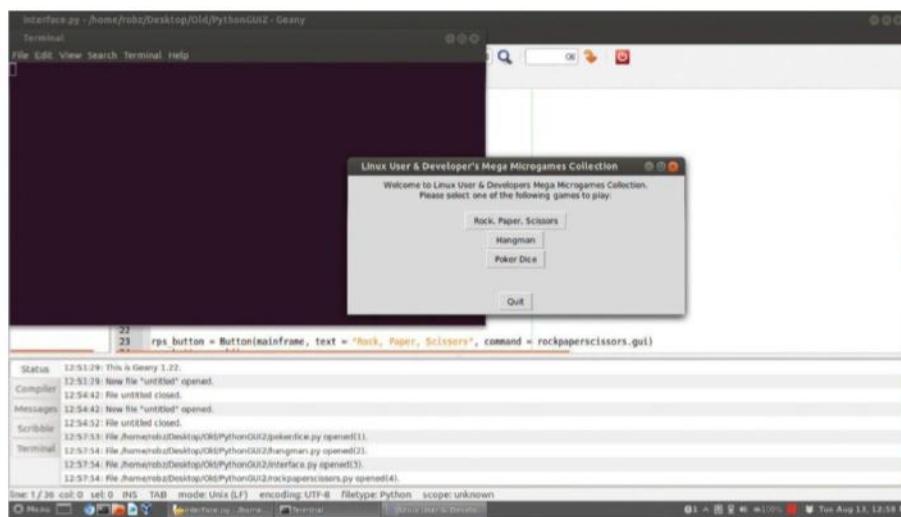
■ Geany's interface is smart and easily breaks down code for workflow purposes

**“Code is automatically highlighted with the correct syntax”**

Sadly, Geany's plug-in support is pretty dire. Some plug-ins extend the basic functionality for a few of the languages, but there's not the kind of depth as Netbeans or Eclipse in the sheer number of available plug-ins.

There are a lot of ways to customise Geany with the standard tools and menus, though. Almost every part of the interface and workflow is editable, with ways to even change the characters required for autocomplete suggestions.

Geany is a very smart IDE, with a low barrier to getting your project started straight away. It does lack in some of the features of some of the bigger IDEs, such as more advanced unit testing and debugging, although it will let you browse compile errors. The plug-in selection is also pretty poor, so it may not be extensible for specific functions.



- The native languages are supported quite well with compilers and such

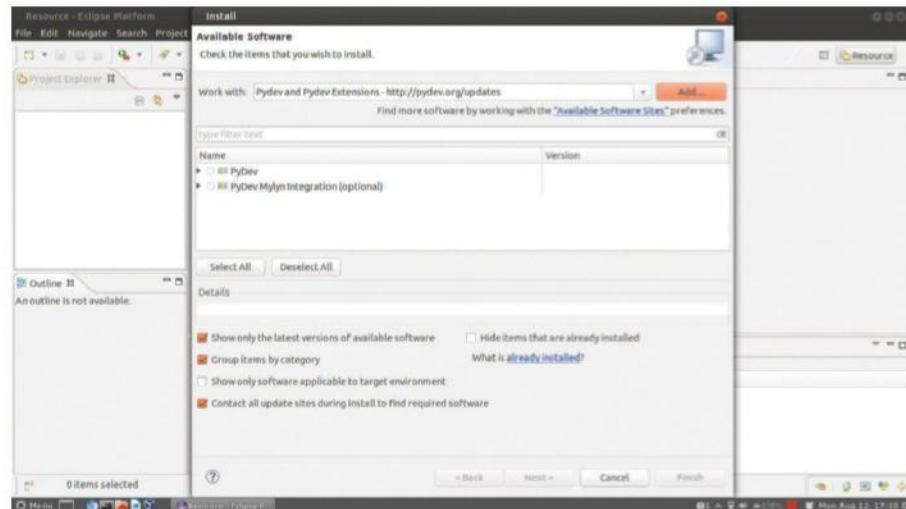
SCORES		
Installation	Readily available in most repos and requires minimal dependencies	9
Workflow	Workflow is superb for the variety of different languages included	9
Features	No proper debugger, but a great selection of build and project management functions	7
Plug-in support	Although it doesn't quite need it like the others, there are very few plug-ins available	4
Overall	Geany is a great, lightweight IDE with good native support for a variety of languages, but there's not much room for expansion	8

# Eclipse

The ubiquitous Eclipse is an industry standard – how does it fare against more community-run efforts?

**Eclipse is one of the most popular development suites around, and at first glance it's easy to see why.** While created mainly for Java development, Eclipse is highly customisable through plug-ins. Thanks to its popularity and community, this has resulted in a great selection of add-ons that enables Eclipse to work with just about any language. These plug-ins allow for more than that, with a marketplace full of interface and behavioural modifications alongside the language elements.

Eclipse has great project management tools as well, with a smart tabbed interface, and plug-in-specific menu entries for starting projects in different IDEs if needed. Projects are located in plain folders in the location of your choice, allowing for easy access of source code if you need it, instead of being inside a proprietary file. The function search ability works well and the interface has plenty of tips and warnings for



■ Plug-in support for Eclipse is top-notch

anything that might be inefficient in terms of the code. There are plenty of editing tools too, such as simple indent or dedent options.

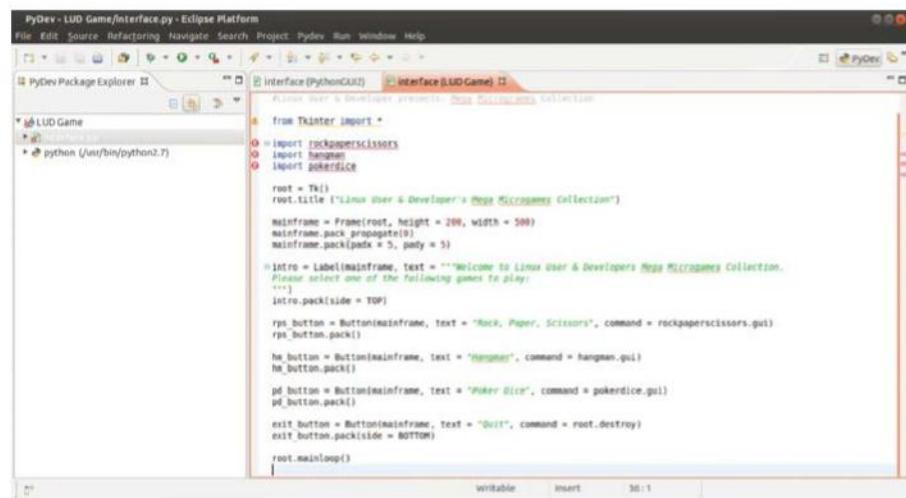
The debug suite in Eclipse is fully featured, with various ways to run, check and unit-test code, although this depends on the plug-ins to some degree. The tools are there, though, and most of the major plug-ins seem to use them.

The plug-ins are handled by a repository system, which lets you keep any add-ons up to date. While there aren't a huge amount available by default, it's easy enough to add more to the plug-in manager and you can even

select which extensions to install from each of the repositories.

Eclipse is customisable in other regards, with an expansive properties and settings menu that lets you edit a huge amount, from the way patches are viewed to little things like key bindings and other shortcuts. Eclipse is a fairly big suite of packages, though, and easily the biggest resource hog out of all of the IDEs we're testing. It does have probably the best extensibility of all these IDEs, however, meaning it also probably has the most to offer those who work on a lot of differing projects.

## "A great selection of add-ons enables it to work with just about any language"



■ Code navigation and highlighting aids your workflow

### SCORES

Installation	Eclipse requires a lot of dependencies; however, it's available in most major repos	7
Workflow	Smart interface design that lets you easily navigate projects and code	9
Features	A great set of features by default, although mainly for handling Java	8
Plug-in support	The best support around, with a great repo system offering a wealth of add-ons	10
Overall	Eclipse is popular for a reason – it's got a great selection of features that are easily extensible to suit almost any need	

7

9

8

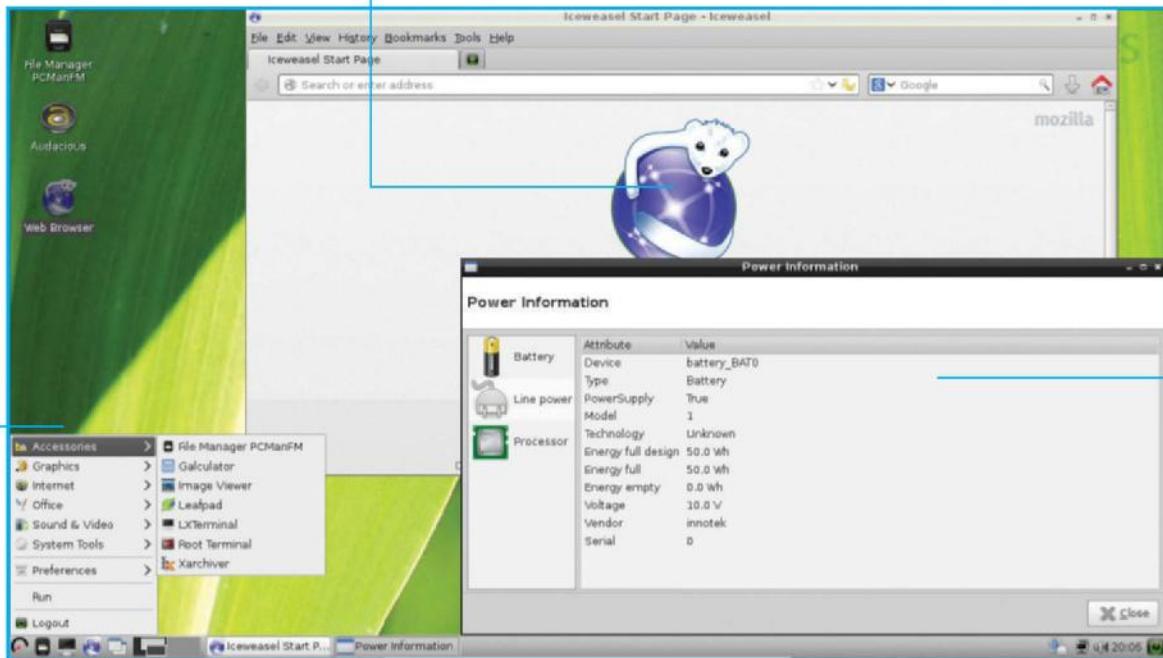
10

9

The selection of apps is extremely low; there's not even an office suite to start with

On the surface, not much has changed since the switch to Debian, however Iceweasel is the most noticeable

wattOS cares about your battery and power usage, and is one of the most energy-efficient distros around



# wattOS R8

The green distro makes a switch from Ubuntu to Debian. Does it make the lightweight distro better or will we return to wattOS 7.5?

## Pros

Lightweight and energy efficient, wattOS is also very easy to use, with few sacrifices made to achieve this.

## Cons

Installation leaves one of the trickiest parts as a manual operation and the image size is still large.

### Lightweight Linux distributions are inherently energy saving.

By definition, you're using fewer resources to run your system, which in turn requires less power and electrical draw. Throw in some power-efficient hardware and idle power draw will be minimal. These lightweight systems – while naturally energy-conserving – don't normally include any specific optimisations for power saving. This is where wattOS comes in.

While also lightweight, wattOS strives to strike a balance between conservative code and usability. The net result is a little less wattage while idle and a longer-lasting laptop battery when disconnected. It's the usability part that is very important to wattOS: something like Puppy Linux or Tiny Core may likely be less resource-intensive while idle, however you need to make some level of sacrifice regarding the desktop and available software to use these distros.

To further its goal, the newest version of wattOS has switched to the current stable branch of Debian 7.0, Wheezy. Before version eight the distro was running on Ubuntu, stripping away many of the core components. With Debian the team can actually build it up more than strip it down, making for a better product overall.

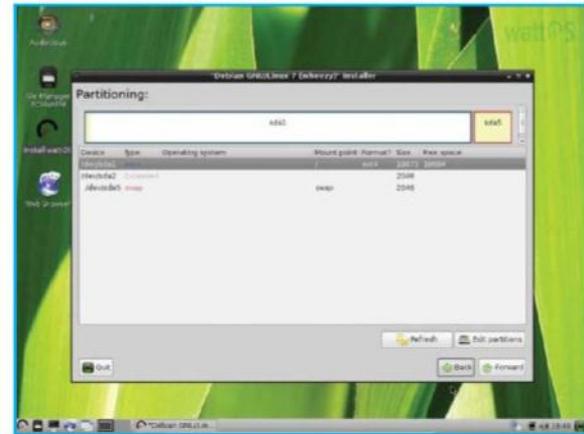
### Debian differences

wattOS still comes in three main flavours: LXDE, MATE and the ultra-slim Microwatt spin that has switched from PekWM to Openbox. Openbox is about as light as PekWM but a little more popular and better supported, resulting in a better overall experience while still providing maximum power benefits.

The LXDE version remains the flagship version of wattOS, sitting in between MATE and Microwatt as a perfect balance



Although the default install comes with a low app count, it can be easily inflated with access to Debian's repos



Installation is generally fine but the manual partitioning seems like the method of yesteryear

## "The result is a little less wattage while idle and a longer-lasting laptop battery when disconnected. It's the usability part that is important to wattOS"

in terms of required resources and usability. The switch to Debian hasn't reduced the size of the images, unfortunately – this was a concern we had with the last round of wattOS releases and the ISOs for R8 are even larger than the ones found in R7.5. This is a minor issue, but one that can be important for lightweight distros.

Installing from these images is not as easy as some of the more major distros. Setting up your user account, time zone and other little things are kept simple, yet you are required to manually partition your hard drive. This is no big deal for a lot of Linux users, but there's no real description or instructions on how the hard drive should be laid out. This can very easily confuse newer users or those used to the ease of Fedora, Ubuntu and other modern browsers; it's an unnecessary barrier to entry in a landscape where excellent and easy-to-use installers are the norm.

### Lightweight software

The main difference you'll notice software-wise with the switch is the use of Iceweasel over Firefox; this is the standard Debian alternative to the quintessential open source browser. The main difference is that, while based on Firefox, it doesn't receive the same level of constant updates and retains an older aesthetic to the overall design. While Midori may be a more traditional choice for a lightweight distro, Iceweasel is much less resource-intensive than the full version of Firefox.

It's flanked by a small selection of other light apps such as Audacious for music, VLC for video and a basic PDF reader. The entire system takes up less than three gigabytes all together, however you have full access to the rest of the standard Debian packages via the Synaptic Package Manager.

All of this allows the distro to boot very fast, even on older hardware. Within seconds we were at the login screen on a more modern system; loading apps and general browsing was fast and smooth, and the memory footprint stayed fairly low relative to other distros. Most importantly, we found no decrease in battery life over our course of testing the distro. Without more thorough testing we'd suggest it was a touch better than with Ubuntu, however it will entirely depend on your workload as well as the load on the system.

### Green fingers

The latest version of wattOS has managed to keep to the high standards of the previous few versions, even with this major shift to Debian. It's not without its gripes, though: the installer should be better and the ISO could probably go on a diet to lose an extra few megabytes.

These issues aside, the presentation, speed and day-to-day product of wattOS is a solid and lightweight distro that truly cares about power usage and the user. If you're looking for a new distro to power an older laptop – or even any laptop in general – this is an excellent option.

### Summary

Some minor issues aside, wattOS does what it sets out to do perfectly. It's stable, fast and generally very easy to use, which is in no small way thanks to the new Debian base. Get it for your laptop now.



**Download now**

[www.planetwatt.com](http://www.planetwatt.com)

# Dropbox

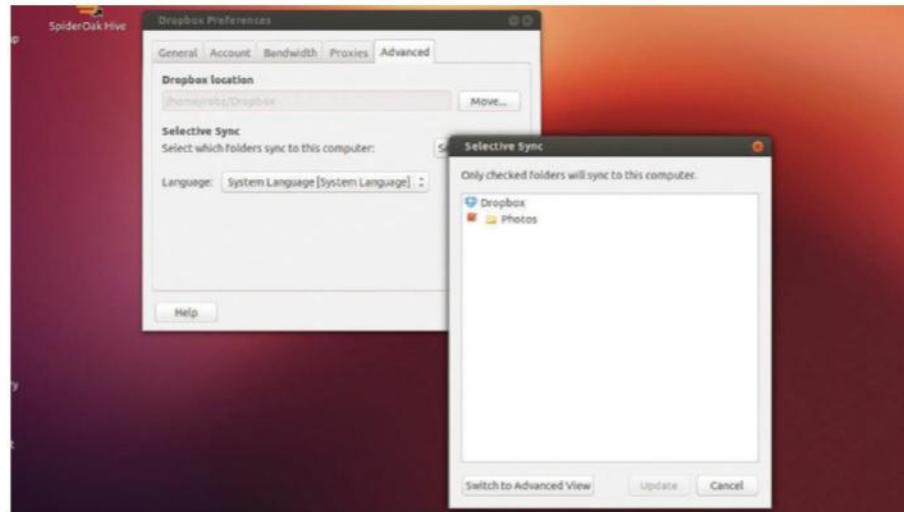
No introductions necessary for the king of cloud storage

Dropbox is synonymous with cloud storage; in fact, when describing other cloud services to less tech-savvy people, the conversation usually includes the phrase ‘it’s like Dropbox’. Popularity doesn’t necessarily mean quality, though, which is why we’re taking an in-depth look at its features to see if it deserves its reverence.

Here are the facts then – Dropbox is a cross-platform service offering 2GB of free data, with small increments available for referrals and such, which is upgradeable for a price. This starts at \$9.99 a month for 100GB, which seems to be somewhat of a standard in the cloud storage space and is cheaper than Ubuntu One’s 20GB add-on scheme. Dropbox’s Linux client is fairly simple compared to the other platforms it works on; however, it offers roughly the same level of functionality without some of the bells and whistles those versions have.

The client only allows for syncing of the main Dropbox folder, rather than letting you select different folders to also sync or back up.

**“Dropbox’s Linux client is fairly simple compared to other platforms”**



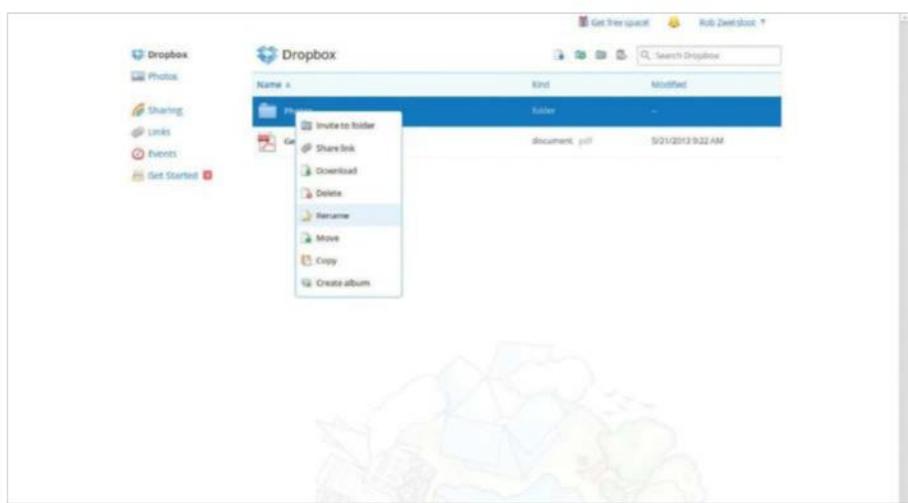
■ Selectively sync specific folders in Dropbox

However, you can tell Dropbox to selectively sync specific folders from the main Dropbox directory, just in case there are files that only need to be accessed by specific devices, or just via the web interface. You can also share folders between other users, allowing you to collaborate on work, or just quickly and easily transfer files between each other. If you also have another client on the same network, it will allow you to transfer files over the LAN automatically – not a huge feature, but good nonetheless in certain situations.

The web interface for Dropbox is also one of the best, allowing you to easily navigate, edit and

change settings on files on your account and in any shared folders that you have ownership of. There’s also a pretty advanced undelete function that lets you browse the last few weeks of files you’ve deleted and restore them as long as you have space in your account.

The good thing about Dropbox is that it’s available on all desktop and mobile platforms, and it works pretty much seamlessly between all of them. While it doesn’t offer all that much space for free, it does have reasonably priced upgrade options and is one of the most stable services available.



■ Have more control over your files via the web interface

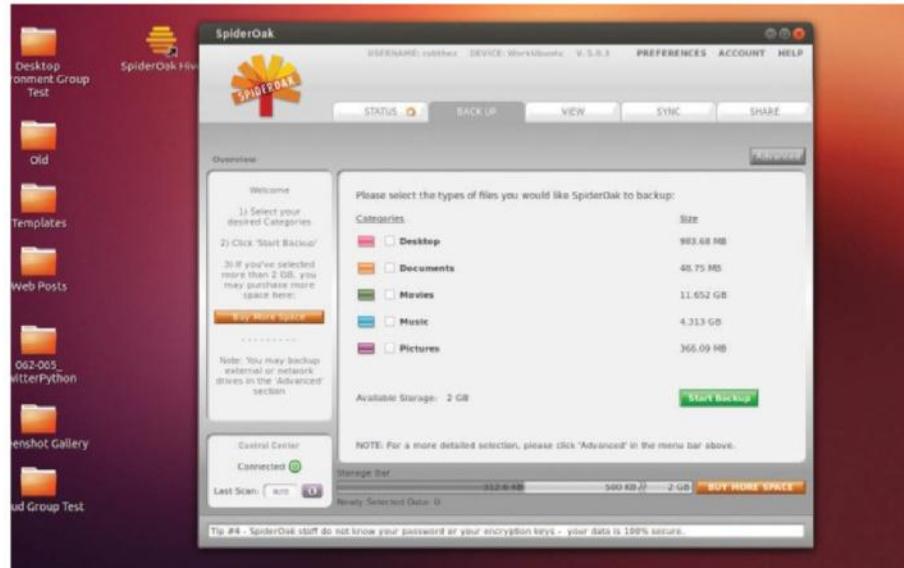
SCORES		
Space	Not a lot of storage space given away for free, but very reasonable upgrade fees	6
Integration	Integrates throughout a network and has a useful notification icon	8
Features	A good set of syncing features and sharing, but no extra folder backup options	7
Customisation	There's actually not a huge amount to change in Dropbox, although there are proxy settings	7
Overall	Dropbox is very good and deserves its reputation. It's solid, reasonably priced and works very well and very quickly	8

# SpiderOak

## A veteran in Linux cloud storage

**One of the first commercial cloud storage services to make its way to Linux, SpiderOak has been around for a few years.** This means a couple of things right off the bat – firstly, it has the same 2GB free, \$9.99 for 100GB storage deal as Dropbox. Secondly, it also has probably the best and most mature client program out of any in this test, which we'll shortly go into more detail about. One of the benefits of using SpiderOak is that the firm behind it is extremely confident in the security and privacy of your data on its server. It operates what it calls a Zero Knowledge scheme, where the server doesn't actually know what you have stored on it because it's encrypted. This is a good idea and should guard against most types of intrusion.

As mentioned, the client for SpiderOak is extremely good on Linux. The main interface supplies a lot of information on the current status of backup and syncing, what the current queue of uploading or downloading files is, a log of any changes made etc. Like Ubuntu One, you can select other folders to back up, as well as having a SpiderOak Hive folder that syncs between all devices. You can also heavily customise the way the client works – such as telling it to only back up files of specific size range, age range or excluding specific keywords;



■ You get complete control over the client and how it syncs

## “The server doesn’t know what you’ve stored on it because it’s encrypted”

setting a backup and syncing schedule in case you don't want it to do it all the time; and it also includes LAN sync like Dropbox.

While the website end isn't as advanced as Dropbox's, there is an Android app which allows you to access your files more easily on the go. One of the neat functions included with SpiderOak, though, is the ability to sync two folders between devices, or on the same device, without having to use the dedicated syncing

folder. This is particularly useful for syncing between folders on the same device if you need some kind of a backup, or multiple users are working on the same system.

Overall, SpiderOak is an extremely mature cloud storage service – and unlike other solutions, it hasn't skimped on the Linux support.

A screenshot of the SpiderOak web interface. At the top, it shows account details: Name: Rob Zwartbos, Account Size: 2 GB, Number of Devices: 1, and Current Backup Size: 419.51 MB. Below this is a 'Folder/File Name' tree view showing 'WorkUbuntu', 'Documents', 'Pictures', and several screenshots from 2013. To the right is a detailed file list table with columns for 'Size' and 'Modified'. The table lists numerous files with their sizes and last modified dates, such as '166.8 KB' on '2013/01/21 4:00:51 PM' and '10.7 KB' on '2013/01/22 5:56:48 PM'.

■ The web interface is basic but usable

### SCORES

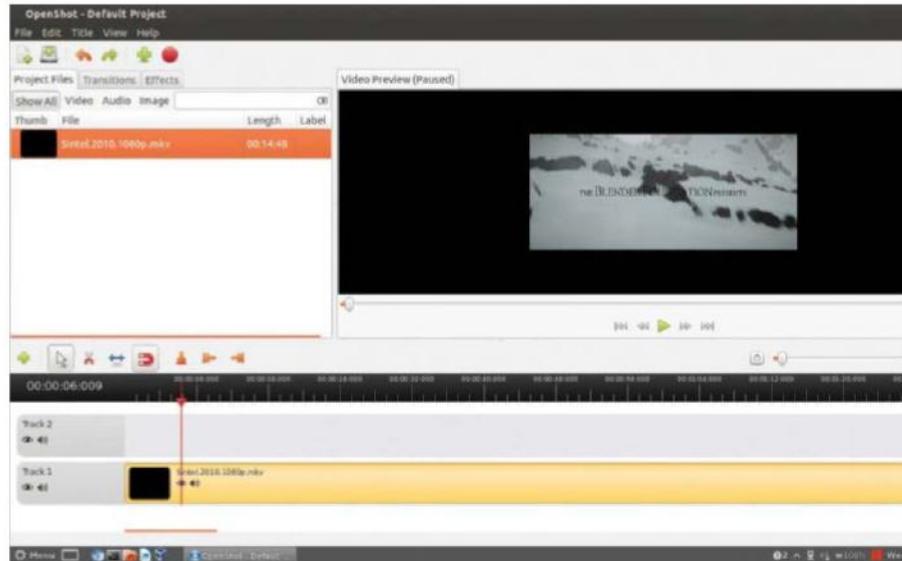
Space	Like Dropbox it only offers 2GB of free storage, but there are decently priced upgrades	6
Integration	SpiderOak is able to integrate well with a number of setups	8
Features	It offers the highest number of features we've seen for a cloud storage service	9
Customisation	Just about every aspect of the client is customisable with useful functions that can be enabled or disabled	9
Overall	SpiderOak is probably the best commercially available storage solution on Linux, and we don't say that lightly	

# OpenShot

An intuitive yet professional movie-making option

Built upon its own media framework, OpenShot is a fairly all-encompassing package that has similar codec support to FFmpeg. It's one of the newer entries in this group test, with its first release just over four years old now, and it's definitely one of the better editing suites we've come across. Like PiTiVi, the interface is nice and straightforward; however, unlike PiTiVi, this interface makes it easier to access a much deeper library of features and editing options.

The timeline consists of a track hierarchy, with higher tracks generally having dominance over the lower tracks in terms of what is previewed and encoded. There's no differentiation between music and video tracks, but it's smart enough to know not to cut off video if an audio track is placed higher than a video track. These tracks can be moved between by the use of transitions, which are visually represented very nicely with an arrow to determine the direction of play, bridging the tracks together to give a nice sense of the flow of the video itself. There are plenty of transitions available, ranging from dissolves,



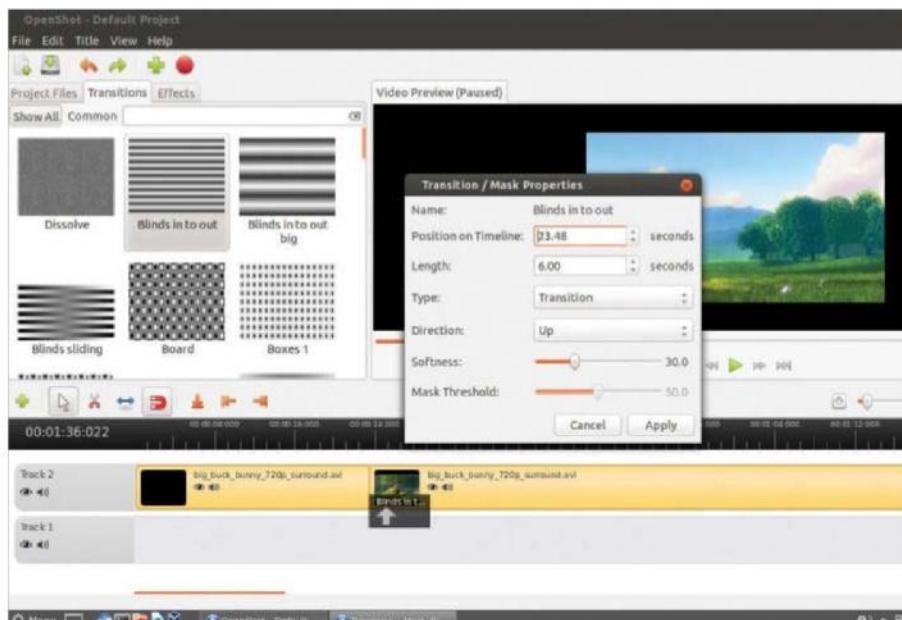
■ The layout is clean yet hides a lot of depth

wipes and fades to either emulating a specific style or even creating some impromptu special effects. There are some visual and audio effects available as well, which are added to the entirety of the clip in the timeline rather than a selected area. In this case you'll need to split up tracks to break up the effects. You can split off the audio from a video track as well if you so wish.

There are a fair number of rendering options, with plenty of presets and more advanced customisation available for bitrate, file format,

aspect ratio, quality and more, so you can create your perfect video file. This goes hand in hand with OpenShot's compatibility with a wide range of codecs: it's able to import from MKV containers as well as lots of other file formats and codec types.

OpenShot's slightly more advanced workflow over PiTiVi is a fantastic yet still easy-to-understand addition, which allows people to easily make much more advanced video clips and movies. Overall, OpenShot manages to combine ease of use with a good feature set.



■ There are a great selection of effects and transitions that are fully customisable

## SCORES

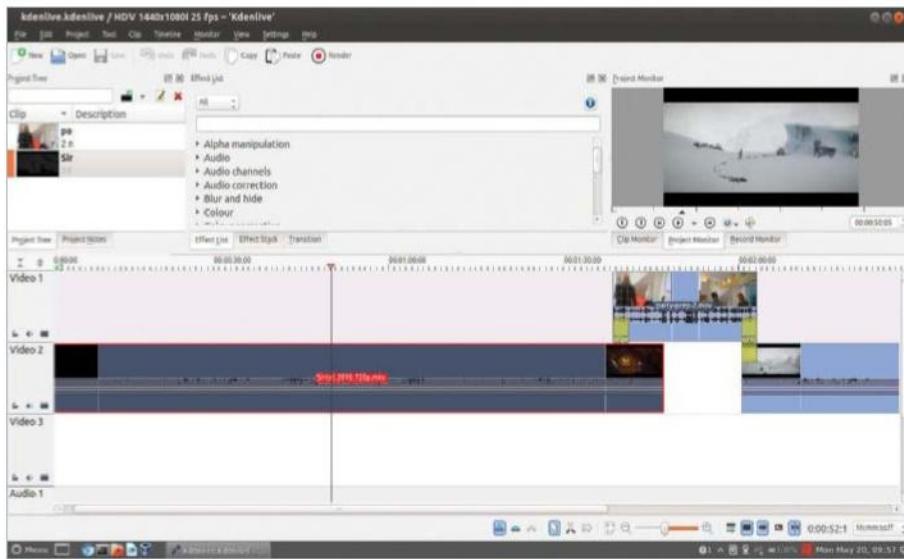
<b>Installation</b>	Widely available in repos, and has its own media framework	<b>9</b>
<b>Ease of use</b>	One of the easiest editors to use in this test, without dumbing down the features	<b>9</b>
<b>Features</b>	A great selection of video editing tools and tricks, as well as effects and transitions	<b>8</b>
<b>Codec support</b>	We found few issues importing video, and there were many export options	<b>8</b>
<b>Overall</b>	A prosumer-level video editor that is only slightly more difficult to use than your basic video editor	<b>8</b>

# Kdenlive

A full-featured video editor – it's the complete package

Whereas OpenShot is a straightforward video editor with a deceptive amount of depth and large number of features, Kdenlive could prove a little more intimidating. With a more utilitarian interface and workflow, Kdenlive is deceptive in its design, however, and generally just as easy to use as OpenShot. The KDE video editor has been updated in recent years to work a lot better on other desktop environments and is also based on the popular FFmpeg media framework, giving it a lot of compatibility with various containers, formats and codecs.

Like OpenShot, Kdenlive works on a hierarchy of separate tracks, with higher tracks having priority, and transitions being used to go between the clips up and down the tree. However, in Kdenlive there is a differentiation between video and audio tracks, allowing you to perform some audio-centric manipulation to the tracks and clips in your timeline. Effects



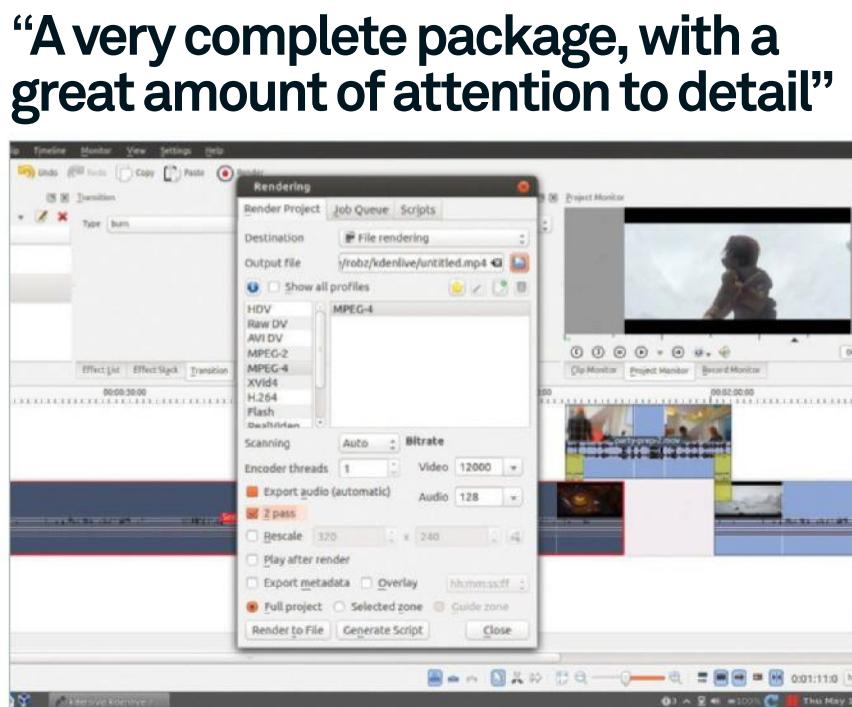
The multi-track interface is very effective

and transitions are generally easier to get to as well, with the right-click menu bringing up a list of both to access. These effects can also be heavily customised to your exact liking.

There's a whole host of other video editing features in Kdenlive to use as well, with a stabilisation algorithm and the ability to create noise tracks, countdowns and clicks

among many others – it's a pretty advanced piece of software. There's also the ability to create a video DVD, taking already DVD/MPEG2 rendered material either from Kdenlive or another DVD to create your own. Rendering itself is fairly quick, with a queue so you can have multiple things ready to go at once, or even transcode other files to use elsewhere.

Kdenlive is a very complete package, with a great amount of attention to detail to a lot of its core and extended features that make sure it does just about everything you'd want it to do. With everything highly customisable, including the presets and encoding profiles, Kdenlive is a fantastic prosumer-level video editor.



Video exporting is highly customisable

## SCORES

<b>Installation</b>	Built on FFmpeg and easily obtainable	<b>8</b>
<b>Ease of use</b>	There is some sacrifice to usability for the sake of features	<b>8</b>
<b>Features</b>	A wealth of options to edit together any video project	<b>10</b>
<b>Codec support</b>	Due to the FFmpeg core, its video support extends far	<b>9</b>
<b>Overall</b>		<b>9</b>
With a little more to offer than most, and decent compromise on the interface for it, Kdenlive is one of the best Linux video editors around		

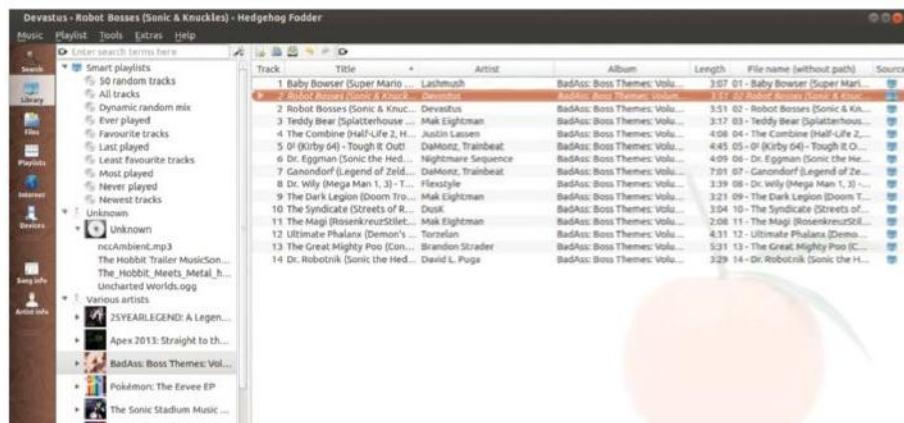
# Clementine

A Linux favourite, how is the latest Clementine player?

Clementine is based on the KDE music player, Amarok, but with a few improvements and a much better interface. It's quickly become a very popular media player and the latest version, 1.2, has arrived with a whole host of great new features. These are sure to attract new users while appeasing die-hard fans who still want to use their favourite media player in a changing landscape of music consumption.

First of all, Clementine now has access to a lot more music streaming services than before, with new additions such as Dropbox and Ubuntu One joining the already impressive list of existing ones. These include Google Drive, Spotify, SoundCloud, Last.fm and Grooveshark. You can easily search within the free services using the built-in Clementine search functions, and you can log in to do the same with the account-driven services such as Spotify and the cloud storage ones. These settings are easily found in the preferences menu under a different section to the vast wealth of customisation options that Clementine offers.

Through these options you can change just about every way Clementine behaves, from simple things like how it might fade between



■ Clementine has everything but the Kitchen Sink. It even has the Hypno Toad

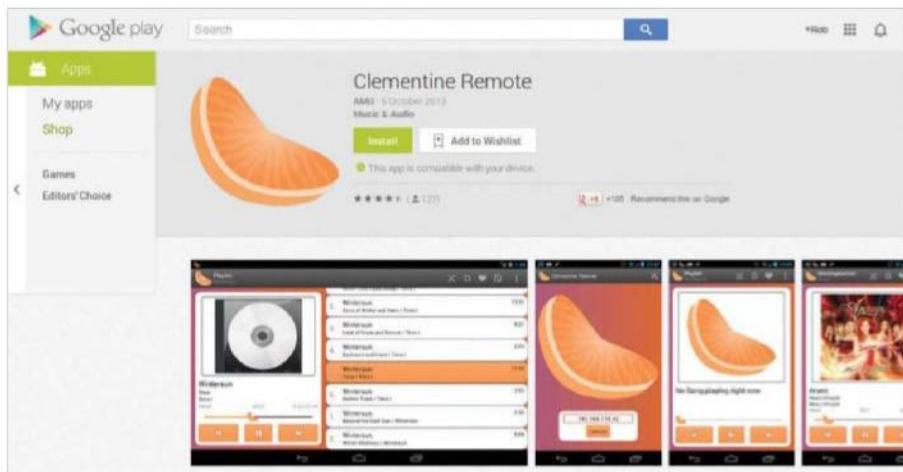
tracks, to tweaking the transcoding settings or even setting a Wii Remote as a remote control device. New in Clementine 1.2 is the ability to use an Android device as a remote, a feature which has been a long time coming. However, instead of using a basic HTTP interface, it uses a special app to make it work.

Playback is fantastic, with a special Clementine icon ticking down to the end of the song, and showing a play symbol so you know it's actually going. While you can control

Clementine from here, you can also control it from the usual volume control icons if you're using the right desktop environment.

Clementine basically has it all, then. Its smart playlist feature, the dynamic random mix, isn't quite as good as some online equivalents, but it's a lot better than any of the other players in this test. It also has the greatest selection of online services it connects to, is the most customisable and makes finding your music easy.

## "You can change just about every way Clementine behaves"



■ The Android app offers better control than some HTTP interfaces

### SCORES

Playback	Makes playback as easy as it can be, short of dedicated buttons on the notification area	9
Interface	The interface contains a lot, but does the best it can for the amount of features	8
Management	Easy to navigate and find media, although some of the online services could work better	9
Online	Connects to everything you would probably want to use bar Pandora and Google Music	10
Overall	An amazing piece of software that lets you do just about anything you'd want to do with all of your music	
	9	

# Banshee

Not as popular as Clementine, but still a great option

Banshee was the one-time default audio player for Ubuntu, replacing and then being usurped by Rhythmbox. Due to this, you would be forgiven in thinking that they're incredibly similar applications – and in some regards they are. They both employ a similar three-pane layout for your media, and they both include a column down the side for navigating your media, videos, podcasts and online services. At the core, they also both run off GStreamer, which is a great media back-end and allows the two to play just about anything with the right codecs installed.

The interface for Banshee is nice and easy to use, and very responsive. Search is instant, bringing up results as you type, and the way results are listed is conducive to finding the tracks, album or artist you're looking for. The album pane on the main interface has thumbnails of the album art instead of a list – although the grid effect can be disabled if you wish. It all works very well and, like all the others, integrates just fine with the desktop



■ The Banshee interface is very nice

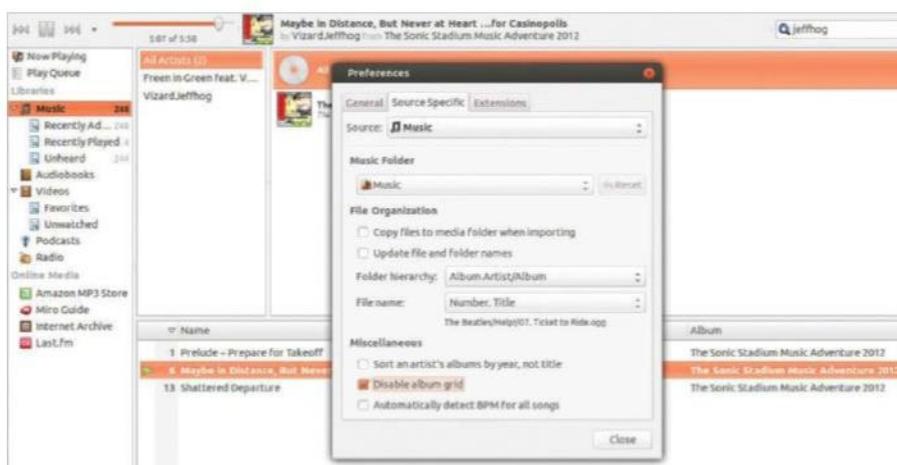
environments that allow for playback options via volume controls.

Customisation wise, there's not a whole lot more than Rhythmbox. You can't even set a specific interval or time for the music library to update. These kind of features are sorely missing, especially compared to Clementine and Audacious which have a whole host of different features and options that can help you streamline the experience. At the very least, there's a fairly rich plug-in system and you can turn off some of the features of Banshee

you don't wish this way, making it much more lightweight than it is by standard. It's through these extensions that the online services are included in Banshee – like Rhythmbox, though, there's only a handful like Last.fm and Amazon. There are a few other, community-built extensions, but none to challenge the features of Clementine.

So overall, Banshee is pretty good. While it's easy to compare it to Rhythmbox, it's generally a little better, with better plug-in support that allows it to be more lightweight if you wish, and a slightly cleaner and informative interface. It's no Clementine, though.

## “Search is instant, bringing up results as you type”



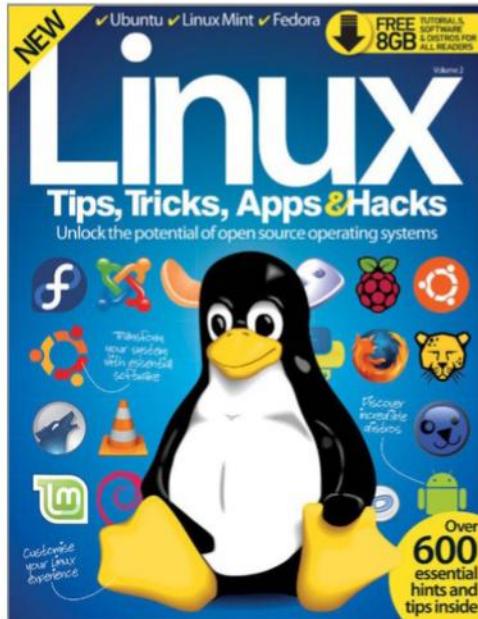
■ Extensions give Banshee a lot of its features, and turning these off is the main way to customise the software

## SCORES

<b>Playback</b>	Good playback options, but lacks its own dedicated notification icon	<b>7</b>
<b>Interface</b>	An easy-to-use interface that is laid out in a logical manner	<b>9</b>
<b>Management</b>	For local content, it's a great way to keep track of all your media of any type	<b>10</b>
<b>Online</b>	Limited online options, and it's only minimally extendable	<b>4</b>
<b>Overall</b>	Banshee is a great media player that we'd be very happy to use if we didn't have access to any online services	<b>8</b>

Special  
trial offer

Enjoyed  
this book?



Exclusive offer for new



Try  
3 issues  
for just  
£5\*

\*This offer entitles new UK direct debit subscribers to receive their first three issues for £5. After these issues, subscribers will then pay £25.15 every six issues. Subscribers can cancel this subscription at any time. New subscriptions will start from the next available issue. Offer code ZGGZIN must be quoted to receive this special subscription price. Direct debit guarantee available on request.

\*\* This is an US subscription offer. The USA issue rate is based on an annual subscription price of £65 for 13 issues which is equivalent to \$102 at the time of writing compared with the newsstand price of \$16.99 for 13 issues being \$220.87. Your subscription will start from the next available issue.



About  
the  
mag



Dedicated to  
all things Linux

**Written for you**

Linux User & Developer is the only  
magazine dedicated to advanced users,  
developers & IT professionals

**In-depth guides & features**

Written by grass-roots developers and  
industry experts

**Free assets every issue**

Four of the hottest distros feature every month –  
log in to FileSilo, download and test them all!

subscribers to...

**LInuxUser**  
**&Developer**<sup>TM</sup>

Try 3 issues for £5 in the UK\*  
or just \$7.85 per issue in the USA\*\*  
(saving 54% off the newsstand price)

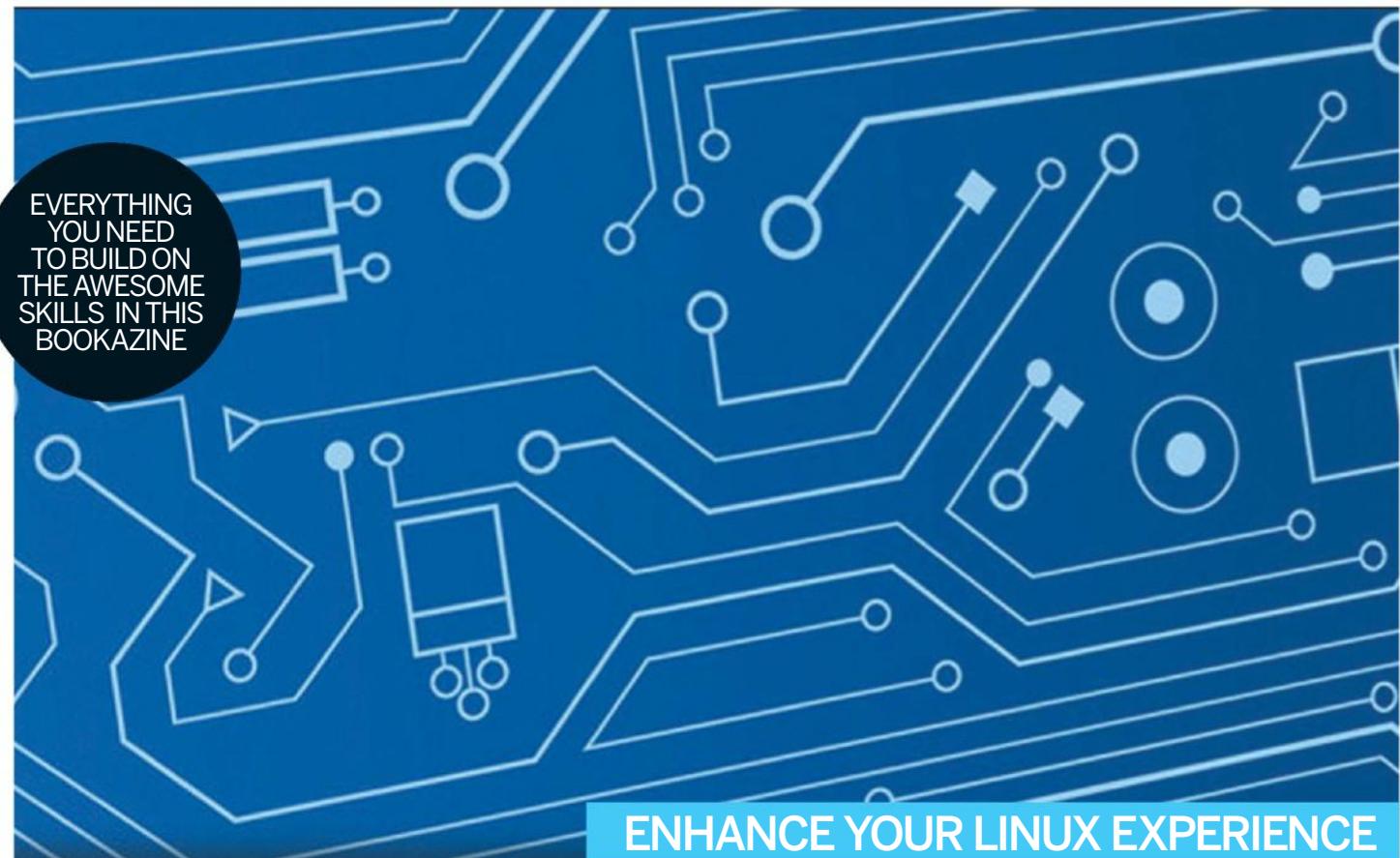
For amazing offers please visit  
**[www.imaginesubs.co.uk/lud](http://www.imaginesubs.co.uk/lud)**

Quote code ZGGZIN

Or telephone UK 0844 249 0282 overseas +44 (0) 1795 418 661

# YOUR FREE RESOURCES

Log in to [filesilo.co.uk/bks-597](http://filesilo.co.uk/bks-597) and download your great resources **NOW!**



## Top-rated distros

PACKED WITH BRILLIANT  
DIGITAL CONTENT, AVAILABLE  
ANY TIME, ON DEMAND

## Recommended software



Clementine



## In depth tutorials

## YOUR BONUS RESOURCES

ON FILESILO WITH THIS  
BOOKAZINE, FREE AND  
EXCLUSIVE FOR LINUX TIPS,  
TRICKS, APPS & HACKS  
READERS, YOU'LL FIND A WEALTH  
OF RESOURCES, INCLUDING...

- A walkthrough on creating and saving data with a MongoDB database
- A tutorial on how to design presentations easily with Hovercraft
- Three highly recommended distros: Linux Mint, openSUSE and Fedora
- All the software you could need to improve your Linux-based projects



[filesilo.co.uk/bks-597](http://filesilo.co.uk/bks-597)

# 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 the following page

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 programming tutorials or video workshops, 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

EVERYTHING YOU NEED TO KNOW ABOUT  
ACCESSING YOUR NEW DIGITAL REPOSITORY



To access FileSilo, please visit [filesilo.co.uk/bks-597](http://filesilo.co.uk/bks-597)

**01** Follow the on-screen instructions to create an account with our secure FileSilo system, log in and unlock the bookazine by answering a simple question about it. You can now access the content for free at any time.



**02** Once you have logged in, you are free to explore the wealth of content available 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](mailto:filesilohelp@imagine-publishing.co.uk).



**DISTRO**  
**fedora**  
Fedora 19  
The latest and greatest from the Fedora project is here (reviewed this...)  
13 0 3

**CLOUD COMPUTING**  
**ownCloud**  
Create your very own cloud using our tutorial, and the fantastic ownCloud.  
25 0 3

**PROGRAMMING**  
**Hovercraft presentation**  
Built upon two powerful tools – impress.js and reStructuredText – Hovercraft gives...  
21 KB 39 12 3

## NEED HELP WITH THE TUTORIALS?

Having trouble with any of the techniques in this bookazine'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 Linux User & Developer and Imagine Bookazines Facebook pages for all your questions, concerns and qualms. There is a friendly community of fellow Linux and Open Source enthusiasts waiting to help you out, as well as regular posts and updates from the team behind Linux User & Developer magazine. Like us today and start chatting!



[facebook.com/ImagineBookazines](http://facebook.com/ImagineBookazines)  
[facebook.com/LinuxUserUK](http://facebook.com/LinuxUserUK)



✓ Ubuntu ✓ Linux Mint ✓ Fedora ✓ Debian ✓ OpenSUSE

# Everything you need to get the most from Linux

Top hints and tips to guide you through the best open source software and operating systems



## Tips

Program with Python and protect your network with step-by-step guides

## Tricks

Maximise the potential of open-source software with comprehensive tutorials

## Apps

Discover top apps and distros that will improve your Linux-based projects

## Hacks

Enhance your Linux experience by calibrating & customising your system

Build your  
own firewall

In-depth guides and  
tutorials written by  
Linux experts



Digital Edition

GreatDigitalMags.com

VOLUME 2 REVISED